

NUTS AND VOLTS

www.nutsvolts.com
March 2016

EVERYTHING FOR ELECTRONICS

Build the **PiBooth** AND TWEET FUN PHOTOS!! AT YOUR NEXT PARTY!!



TIPS ON NIXING NOISE
from your circuit designs

LED CHOREOGRAPHY
make your LEDs dance

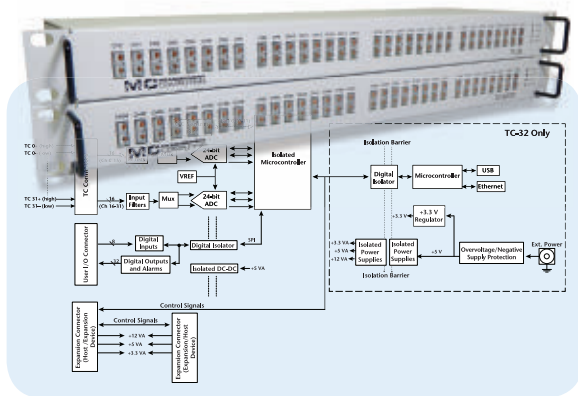
REFLECTING ON TRANSMISSION LINES



TWO GREAT DAQ BRANDS NOW ONE GREAT COMPANY

Easy to Use • Easy to Integrate • Easy to Support

GENERAL PURPOSE DAQ

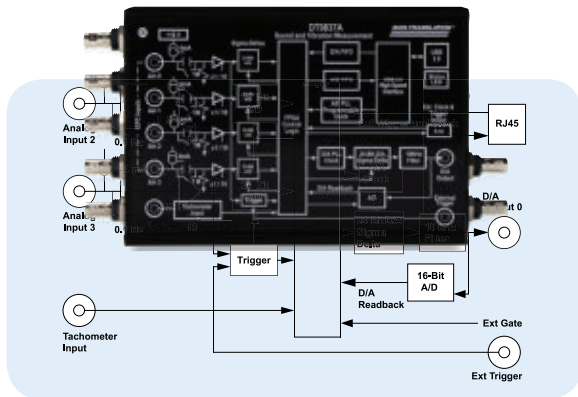


High-Channel Temperature

TC-32 – Key Highlights

- Up to 64 thermocouple inputs
- USB and Ethernet interfaces
- 24-bit resolution
- Per channel out-of-range alarm configuration
- Cold-junction compensation (CJC) sensor per channel
- Open thermocouple detection
- Up to 16 digital inputs, 64 outputs/alarms
- 19" rack mountable
- Included software and drivers

SPECIALIZED DAQ



Sound and Vibration

DT9837 Series – Key Highlights

- 4 Simultaneous, 24-bit A/D's
- USB interface
- Direct IEPE inputs (4 mA current source)
- Up to 105.4 kHz per channel
- Tachometer input
- One 24-bit D/A channel, up to 96 kHz
- Sync multiple modules for expansion
- OEM or BNC packaging
- Included software and drivers

For Sales and Support Contact:



www.mccdaq.com

1.800.234.4232



www.datatranslation.com

1.800.525.8528

Now is the Time to Dive into C



“Why use a PIC® MCU?”

A PIC® microcontroller allows for complicated electronic designs to be simplified with low-cost, low-power and with a small physical footprint.

PIC® MCU hardware design the leading microprocessors for new product designs.



“How do I begin?”

Choose the CCS C Compiler - the easiest to use for the most popular programming language for PIC® MCUs. Select from dozens of ready-to-use CCS Development Kits.

Easy & Affordable

C Workshop Compiler



\$99

Full IDE compiler toolset for four popular devices

E3mini Board & Text Book Bundle



\$85

Best way to learn C

EZ App Lynx Development Kit



Special Offer \$199

Use a PIC MCU to control data displayed on a smart device



www.ccsinfo.com/BeginC

sales@ccsinfo.com
support@ccsinfo.com

Ph: 262-522-6500 x35

PIC® MCU is a registered trademark of Microchip Technology Inc.

26 PiBooth — A Raspberry Pi Based Photo Booth

Photo booths have always been a lot of fun. Now that there are digital cameras available, we can take photo booths up to a whole new level using a Raspberry Pi and a handful of other parts. Plus, this one is portable, so it's easy to add some high entertainment value to your next event no matter where it is.

■ By John Leeman

34 Techniques to Improve Your Signal-to-Noise Ratio

We all want to avoid noise, which is basically an unwanted signal. What do you do, then, if your output level or source signal is at a fixed volume or strength and can't go any stronger? Discover some simple techniques that help you deal with this issue and improve the signal-to-noise ratio in your projects.

■ By Beau Schwabe

08 TechKnowledgey 2016 Events, Advances, and News

Some of this month's topics include becoming a GMO, faxing for free, cheap meters, and the value of an Apple 1.

12 Q&A

Reader Questions Answered Here

This time around, Commodore 64 video problems are addressed, plus a reliability tutorial on electronic devices is included.

18 The Spin Zone

Adventures in Propeller Programming

More Hot & Sticky

If you haven't explored humidity and temperature measurement due to the higher cost of precision sensors, the DHT11 is an inexpensive gateway to a lot of fun.

50 The Design Cycle

Advanced Techniques for Design Engineers

JumpStart Your Coding for ARMs

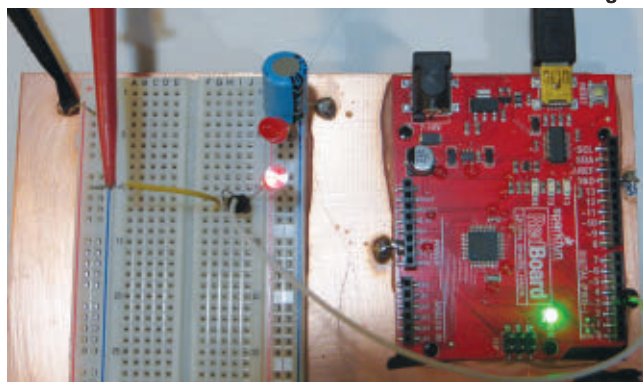
Have you been wanting to start coding for ARM microcontrollers and found that you would have to give up an ARM and a leg in the learning curve? Well, you can keep all of your limbs as you'll need them to immediately start coding for ARM microcontrollers with ImageCrafts' JumpStart API.

38 Make Your LED Dance Just Like a Candle

Looking to add some dynamic lighting to your room? Then, modulate an LED to respond to the world around it with this simple technique.

■ By Eric Bogatin

Page 38



57 The Ham's Wireless Workbench

Practical Technology from the Ham World

Impedance Matching

When RF energy travels in a transmission line and encounters a load or different type of line, the resulting reflections create standing waves of voltage and current, just as water waves do when they encounter an obstruction. This can create problems with digital signals and can result in extra losses. However, there are several techniques for dealing with these reflections.

NEW!

56 Name That Part!

Think you know your stuff? Well, try your hand at our new quarterly feature — a quiz that will test your memory and your wits.

■ By David Goodsell

Departments

05 DEVELOPING PERSPECTIVES

When All You Have is a Hammer

06 READER FEEDBACK

24 NEW PRODUCTS

25 SHOWCASE

46 ELECTRO-NET

48 NV WEBSTORE

62 TECH FORUM

64 CLASSIFIEDS

65 AD INDEX

Published Monthly By
T & L Publications, Inc.

430 Princeland Ct.
Corona, CA 92879-1300
(951) 371-8497

FAX **(951) 371-3052**

Webstore orders only **1-800-783-4624**
www.nutsvolts.com

Subscription Orders

Toll Free **1-877-525-2539**

Outside US **1-818-487-4545**

P.O. Box 15277
North Hollywood, CA 91615

FOUNDER

Jack Lemieux

PUBLISHER

Larry Lemieux
publisher@nutsvolts.com

ASSOCIATE PUBLISHER/ ADVERTISING SALES

Robin Lemieux
robin@nutsvolts.com

EDITOR

Bryan Bergeron
techedit-nutsvolts@yahoo.com

VP OF OPERATIONS

Vern Graner
vern@nutsvolts.com

CONTRIBUTING EDITORS

Fred Eady	Tim Brown
Jon McPhalen	Jeff Eckert
Ward Silver	David Goodsell
Beau Schwabe	Eric Bogatin
John Leeman	

CIRCULATION DEPARTMENT

subscribe@nutsvolts.com

SHOW COORDINATOR

Audrey Lemieux

WEBSTORE MARKETING

Brian Kirkpatrick
sales@nutsvolts.com

WEBSTORE MANAGER

Sean Lemieux
sean@nutsvolts.com

ADMINISTRATIVE STAFF

Re Gandara

Copyright © 2016 by T & L Publications, Inc.

All Rights Reserved

All advertising is subject to publisher's approval. We are not responsible for mistakes, misprints, or typographical errors. *Nuts & Volts Magazine* assumes no responsibility for the availability or condition of advertised items or for the honesty of the advertiser. The publisher makes no claims for the legality of any item advertised in *Nuts & Volts*. This is the sole responsibility of the advertiser. Advertisers and their agencies agree to indemnify and protect the publisher from any and all claims, action, or expense arising from advertising placed in *Nuts & Volts*. Please send all editorial correspondence, UPS, overnight mail, and artwork to: **430 Princeland Court, Corona, CA 92879**.

DEVELOPING PERSPECTIVES

by
**Bryan
Bergeron,
Editor**

When All You Have is a Hammer

A friend in the marketing business contacted me about a project for a local retail store. He wanted to track customer satisfaction as customers exited the store by placing an Arduino-controlled survey taker. Customers would press one of five buttons as they left, indicating their experience from Very Satisfied to Not Satisfied. My friend envisioned five buttons connected to an Arduino, an LCD display, perhaps a beeper for button press feedback, and a battery pack capable of supporting the device for a week.

Seemed like a simple enough task. Too simple, in fact. After working up a straightforward program and defining the base hardware, we naturally progressed to planning a Wi-Fi interface so the counter could be accessed and reset remotely. That would require a simple web page, and maybe a couple hours of programming. We even evaluated a solar powered charger to obviate the need for a plug-in charger.

With plans in hand, we stood back, looked at the hardware and software involved, and the total cost. Then, we revisited the requirements. After a sanity check, we decided the complex Arduino-enabled survey device was overkill.

Starting over without a preconceived product, we identified a solution of five digital mechanical tally counters (or clickers) sold for coaches. We found suitable counters ranging from \$2 to \$5 each on Amazon. The counters — each the size of a walnut — easily fit on a plastic face plate with cutouts for each counter. And it worked. No

batteries to worry about. No programming. And fully reusable counters once the survey was finished. Sure, there was no web interface and no way to check the tally at home on a smartphone, but there wasn't a need.

The take-away from my experience was to avoid preconceived solutions to new problems. Sometimes expertise in one area unnecessarily narrows the range of options that should be considered when assessing a problem.

The caveat, of course, is that you shouldn't pass up a chance to learn and expand your skill set. If your goal is to learn to work with an Arduino or other microcontroller and you have the time and funds, then go for it. Given the challenge above, why not have that Wi-Fi interface? Or, automatic cloud upload?

Go wild with the web interface, with visual and audible alarms, and graphics. Just don't lose touch with what features and functions are really required. Experimenting is fantastic, but know when and how to apply your skills to practical problem solving. **NV**

READER FEEDBACK

What's (not) in a Name?

So, the typo gremlins were out in full force for the January 2016 magazine. In regards to the Keytar article, unfortunately, the writer's last name was misspelled as "Levin." The correct spelling is "Lavin." We apologize for this!

Electronics Footprints

Bryan Bergeron's recent editorial — "For Learning, Old School Can be the Best School" — really resonated in large measure with my own experience. It started with building a crystal set in 7th grade. I was awed by how much pleasure I had tuning in far off radio stations on a clear day; how I could improve reception by learning about circuits. All wire and parts then were from junk radios and coil forms from around the house — all for very little cost. This then expanded into understanding circuits for amplification of sound and the enjoyment of high fidelity. Many hours were devoted to experimenting with different amplifying circuits, speakers, and their enclosures.

As I reflect, it started with the Edisonian approach to learning, and then getting more academic with the study of physics in college. That early trial and error experience, however, heightened my passion for learning diverse things — in my case, to improve the audio experience so that classical music is exhilarating.

I support the premise of your article — especially for the passion that can inspire the young. My observation of many children today is that they have little time to dream about their future because they are driven into sports. I have 11

grandchildren from 12-20, and only one had a technical interest when young; he is now studying mechanical engineering. The others seemed consumed with sports. The technical career path was rewarding for me on several levels, and I hope parents and educators will encourage more young people to follow their technical interests.

R. Straight

Thanks for the note. I've seen the sports focus you refer to. I enjoy sports and think it's important for students to experience teams, but to the exclusion of all else is a mistake for which we're paying the price. I'm glad that your grandson is following your footsteps into a technical future.

Bryan Bergeron

Radio Blast

I've been having a blast with the SDR receiver ("An Ultra Modern Shortwave Radio," by George Steber) in the July 2015 issue. I'm a Ubuntu Linux user, so I found Gqrx as an alternative to SDR# and from there discovered Fldigi, Pulseaudio, and Websdr.

Based on inspiration from Ron Hackett's articles, I did a strip board layout for the 24 MHz upconverter. Amazingly enough, it worked the first time! I did discover that what looks good on a paper strip board layout can be very tight when actually soldering the components, however. I will spread things out a bit more on my next layout. I used Fritzing to print out a bare strip board format and used this to do a component side layout on paper. I then went back into Fritzing to enter the trace cuts and exported this as a png file. I read this png file into Gimp and flipped it side for side to produce a solder side image that I could use to cut the traces on the strip board. I used a 3/32 drill bit in a battery screwdriver to cut the pads and a ball bit in a Dremel to clean up the cuts.

I built the BPF version and I can pick up SWL stations across the HF bands. I have not been able to pull in any ham band activity (as was mentioned in the article). I'm going to add the tuned filter/pre-amp circuit next; one for 80/40/30 meters and one for 30/20/15/10 meters. Hopefully, I will be able to receive the ham bands at that point. Many of the WebSDR servers use RTL sticks and they work fine.

James Lynes

Glad you put the article to good use!

Bryan Bergeron

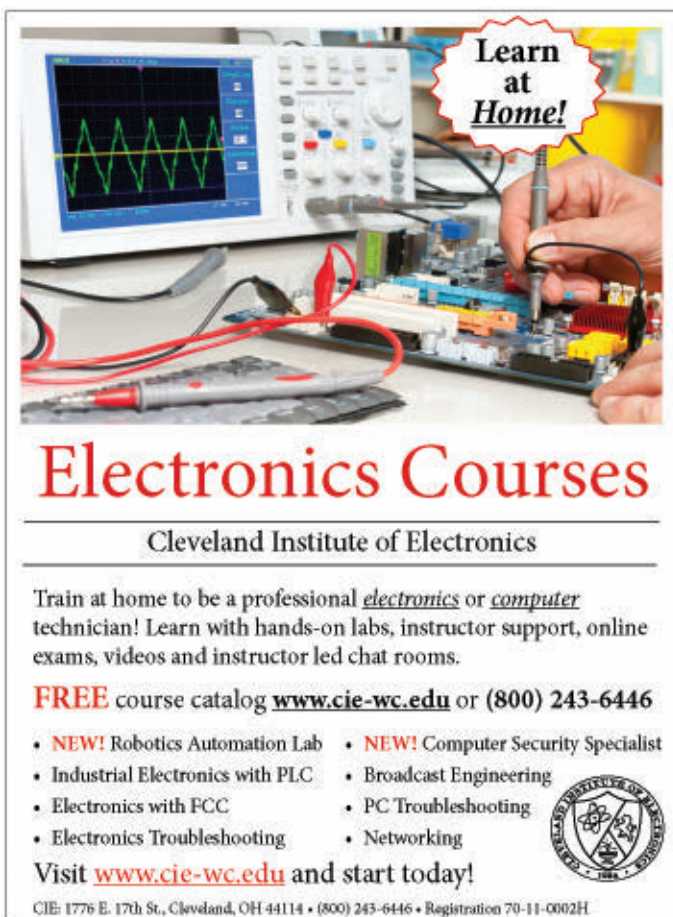
Watch Men

We're both horologists! Can you imagine that?

I read in your editorial in the November 2015 issue that you had an unfortunate encounter with a "magnetized" timepiece. Ugh! I hate it when that happens. Typically, I take everything apart and degauss the culprit part. That part is the hairspring — the only part made of steel inside a quality watch. It's always the hairspring. Even inexpensive watches with steel mainsprings; it's the hairspring that gets all of the "attraction" attention. I usually remove it and run it through the coil treatment (gently pulling it away from my homebrew circuit). Then, back it goes into the watch.

Why does it make the watch run erratic? Simple. As you know, the hairspring drives the balance which toggles the lever; that strikes the pallets, moves the escape wheel, and

Continued on page 64



Learn at Home!

Electronics Courses

Cleveland Institute of Electronics

Train at home to be a professional electronics or computer technician! Learn with hands-on labs, instructor support, online exams, videos and instructor led chat rooms.


FREE course catalog www.cie-wc.edu or (800) 243-6446

- **NEW!** Robotics Automation Lab
- **NEW!** Computer Security Specialist
- Industrial Electronics with PLC
- Broadcast Engineering
- Electronics with FCC
- PC Troubleshooting
- Electronics Troubleshooting
- Networking

Visit www.cie-wc.edu and start today!

CIE: 1776 E. 17th St., Cleveland, OH 44114 • (800) 243-6446 • Registration 70-11-0002H

Nuts & Volts 12 CD-ROMs & Hat Special!

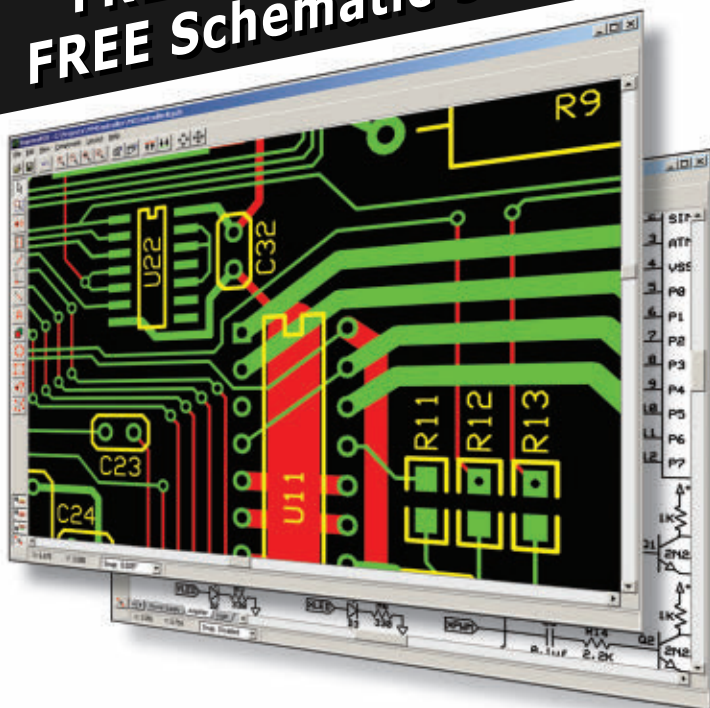


Price \$249.95
Plus FREE Shipping

Call 1 800-783-4624 or visit store.nutsvolts.com

\$51^{For 3} PCBs


FREE Layout Software! FREE Schematic Software!



- 01 DOWNLOAD our free CAD software
- 02 DESIGN your two or four layer PC board
- 03 SEND us your design with just a click
- 04 RECEIVE top quality boards in just days

expresspcb.com

PoLabs

For more information or software download please visit 

www.poscope.com/nv

PoKeys Connect, Control



PoKeys is a succesful story about versatile and extremely powerful controller.

Save your time with fast connectivity and easy configuration of advanced features.

Enjoy rare moments in life while PoKeys is taking care over your house or garden.

Take a break while PoKeys is controlling stepper motors in your CNC or 3D printer.

Let PoKeys make your imagination come true using our intuitive software and plugins for Mach3/4

Stepper motor drivers Drive

Safest and most efficient force for your motors.



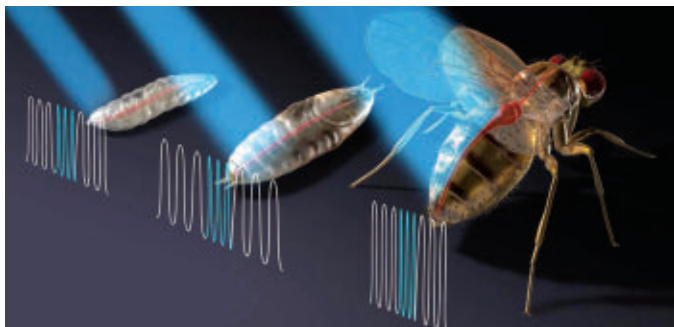
PoScope Mega1+ Measure



Do you need a really powerful and portable oscilloscope that will save your time, money and make your life with measurements easier?

PoScopeMega1+ is lowest power consumption USB oscilloscope, function generator and logic analyzer all-in-one available on the market.

ADVANCED TECHNOLOGY



■ Heart rhythm of fruit flies, regulated by light.

Have a Change of Heart; Become a GMO!

Cardiac pacemakers have come a long way since 1950 when engineer, John Hopps designed and built an external pacing device that was driven by vacuum tubes and powered from an AC wall socket. Not only was the device painful to the patient, it offered the possibility of electrocution as a bonus. Today, most pacemakers are implantable, about the size of two or three 50 cent pieces, and powered by lithium batteries. Still, modern devices have some negative qualities, including the need for implantation surgery, possible damage to surrounding tissue, and an average battery life of only about seven years. The newest versions (not yet approved for commercial use in the USA) have been miniaturized to the size of a large vitamin tablet and are placed directly into the heart via a catheter inserted in the femoral vein. However, some of the same risks remain.

To avoid such issues, researchers at Lehigh University (www1.lehigh.edu) are working on a new approach which is based on optogenetics (i.e., using light to control living cells). The fundamental principle is based on modifying the genes in heart muscle cells so they react with light, and stimulating them with light to set the desired pace. So far, the research has involved fruit flies instead of people — a valid approach given that the two species share about 75 percent of genes that are related to heart disease.

First, they modified the hearts with a rhodopsin (similar to the photoreceptors in green algae). They then stimulated the heart tissue using a laser to penetrate the bug's thin body walls. In all three of their life stages (larva, pupa, and adult), the device worked as designed and caused no damage.

Transferring the technology to humans presents some problems, including how to shine light into our much thicker chest cavity and the need to genetically modify our hearts. If you're terrified of a genetically modified potato, you probably won't want to become a GMO yourself. The developers remain optimistic, however, that eventually a way will be found to use the technique (possibly using infrared light) in people.

Motor Boosts Efficiency

Electric motors don't change much — especially when it comes to evaporator fan motors such as used in the commercial refrigeration industry. Most of them are based on the AC induction (shaded pole) motor, first demonstrated in the 1880s. Then, in the 1960s, someone finally noticed that induction motors are only about 20 percent energy efficient. This led to the development of the electronically commutated motor (ECM) as an alternative, and they are about 60 percent efficient. This is a major improvement but — because a lot of wasted energy is still lost in the form of heat — a system is still constantly fighting with itself.

Fortunately, a new design was announced last September, developed by inventor Joe Flynn and QM Power (www.qmpower.com), located just outside of Kansas City, MO. Dubbed the Q-Sync Smart Synchronous Motor, “Q-Sync’s innovative controller quickly and efficiently eliminates the need for power conversion after the motor starts. Q-Sync’s advanced electronics quickly get the motor to its targeted speed and then efficiently shift the motor to AC power supplied directly from the grid. And because Q-Sync is a permanent magnet motor, it eliminates the additional energy requirements and slip that is associated with a shaded pole motor. The result is a motor that provides up to an 80 percent reduction of energy at the meter and an 80 percent reduction at the grid.”



■ Inventor Joe Flynn with a pair of Q-Sync motors.

Okay, sure, that sounds like the usual marketing hype, but the Department of Energy has decided that the Q-Sync deserves to be categorized as a “High-Impact Technology” and intends to work with the company, OEMs, supermarkets, contractors, and utilities to install more than 10,000 of them in 50+ grocery sites over the next year. A recently released report by the Oak Ridge National Laboratory concluded that replacing existing motors could save businesses \$517 million a year in energy costs, reduce demand on the power grid by around 600 megawatts, and cut annual carbon dioxide emissions by four million tons. Plus, by expanding Q-Sync implementation into other areas, energy consumption could be reduced by at least 300 billion kilowatt-hours. (To download a copy of the ORNL report, go to info.ornl.gov/sites/publications/Files/Pub58600.pdf.)

▲

COMPUTERS and NETWORKING

Gaming Notebook from HP

When you think of gaming computers, one of the least likely names that comes to mind is Hewlett-Packard. But lo and behold, HP (www.hp.com) has introduced the HP Pavilion Gaming Notebook, designed “to deliver intense design, graphics, power, and performance gamers crave in an affordable Windows 10 gaming PC.”

It doesn't look much different from other notebooks except for the backlit keyboard that glows green and the “fierce gradient reptilian pattern on the keyboard deck.” Inside, you get a NVIDIA® GeForce® GTX 950M and a sixth-generation Intel® Core i5 or i7 processor. Also available is up to 16 GB of system memory, 2 TB of hard drive storage, 1 TB of hybrid HDD, or dual storage with a 2 TB HD and a 128 GB SSD. For I/O, you get two USB 3.0 ports, one USB 2.0 port, one HDMI port, and an SD card reader to expand storage.

It sports an optical disk drive, and audio from Bang and Olufsen. The list price starts at \$899.99. ▲



■ HP's Pavilion Gaming Notebook.

Four-Stream Router Boosts Rates

If your Wi-Fi connection is slowing you down, you may be interested in the latest router from TP-LINK (www.tp-link.us): the Archer C2600. Based on 802.11ac technology (backwards



compatible with 802.11b/g/n), it features multiple input/multiple output (MIMO) data transfer and four-stream capability to optimize transfer rates for multiple devices on the network.

According to the company, its 2.4 GHz and 5 GHz bands can reach maximum speeds of 800 Mbps and 1,733 Mbps, respectively. In addition, it employs a 1.4 GHz dual-core processor to boost performance, plus beamforming technology to improve Wi-Fi coverage via stronger, more reliable connections.

Other features include two USB 3.0 ports to share photos, files, and so on; a free tether app to manage your network from your mobile device; and Gigabit Ethernet ports for wired connections. The C2600 will set you back \$249.99 list. ▲

Fax for Free

Most of us tend to keep up with modern technology, so we killed our landlines and dropped off our fax machines at the Salvation Army a long time ago. Years later, though, such entities as doctor's offices, banks, lawyers, and other slow learners cling to their tech dinosaurs and expect us to do the same. In case you haven't noticed, you no longer have to schlep down to the local copy shop and pay them to send a fax for you. Several websites now allow you to fax documents directly from your computer for little or no money.

One of these is faxZERO (www.faxzero.com). All you have to do is log on, enter sender and receiver information, and upload a .doc, .docx, or .pdf file. The site allows you to send three faxes of up to three pages each per day. The only catch is that the company adds a self-promotional image to each cover page. If that's a little too tacky for you, another option is to pay \$1.99 for a commercial-free fax, and this one can be up to 25 pages.

Before you do that, consider that for \$2.99, you can send up to 500 pages per month. Any way you look at it, it's a pretty good deal. ▲

CIRCUITS and DEVICES

Ersatz Segway

Maybe you've always been intrigued by Segway personal vehicles but balked at the \$6,000-\$8,000 price tag. After all, you can get a pretty nice motorcycle for that. So, what if you could get more or less the same experience for less than \$300?

Sold under a variety of brand names at a wide range of prices, we located the Two Wheel Smart Self-Balancing Scooter for \$298 — including shipping — at Technology Pep

(www.technologypep.com). It appears to be identical to the OutTop Two Wheels Self-Balancing Mini Smart Electric Scooter Unicycle by HKCUBE, the RioRand Two Wheels Smart Self-Balancing Scooters Electric Drifting Board Personal Adult Transporter with LED Light, and many others.

It looks kind of junky, but reviewers generally give it four out of five stars, so who knows? Specs include a



■ One of many cheap self-balancing scooters.

maximum speed of 10-12 mph (15-20 km/h), aluminum construction, up to a 12 mile (20 km) range, and a charging time of 1-2 hr. If you don't weigh more than 255 lb (120 kg), it can handle the load. A battery charger is included. ▲

Projector Brings Movie Night Home

Many of us have dusty memories of nights when neighbors or family would drop by with a slide projector, and we'd turn off the lights and chew stale Jiffy Pop while our eyes glazed over to a never-ending procession of images: Uncle Homer sitting on a plastic elephant at Coney Island; Aunt Beulah sticking her tongue out in front of Ruby Falls; or maybe cousin Orville unwrapping a Mr. Potato Head under the Christmas tree. By the end of the evening, we were either nodding off or slitting our own throats.

Now, you can bring back those days on a much less repugnant level. With the Epson (www.epson.com) Home Cinema 640, you can enjoy your favorite TV shows and movies, displaying images up to 300 inches. Plus, with its digital HDMI port, you can view content from a cable or satellite box, Blu-Ray® player, gaming console, or streaming device.



■ Epson's Home Cinema 640, portable and brighter.

The box offers four built-in special color modes optimized to display the best possible quality and color in your particular environment. Additionally, it offers built-in sound and vertical (automatic) and horizontal (manual) image correction. The Home Cinema puts out 3,200 lumens of color brightness and 3,200 lumens of white brightness. The unit lists for \$359.99, which won't even buy you a 50 inch flat screen at Walmart. ▲

CIRCUITS and DEVICES Continued

Need a Cheap Meter?

Once in a while something comes along that is so cheap it's almost free, and the Tekpower Handy Man's Tool DT830B pretty much qualifies. Aimed at hobby and DIY users, the DMM provides a 3.5 digit LCD display with a maximum reading of 1999, built-in diode test, HEF measurement, and overload protection. Plus, it's only \$5. Yep, not a typo.

You may as well buy a dozen just so you'll never have to root around for one again. It's available at www.tekpower.us.



■ The DT830B DMM: cheaper than dirt.

INDUSTRY and the PROFESSION

Apple 1 Worth More at 40

Many years ago, an LSD-soaked hippy sold his VW Microbus, an electronics engineer sold his HP-65 calculator, and a third individual sold his interest in the three-way partnership for \$800 and bailed. We're talking about Steve Jobs, Steve Wozniak, and Ronald Wayne. The result was the Apple I computer, introduced in April of 1976. (Had Ron kept his Apple stock until 2015, it would have been worth about \$60 billion.)

Yes, it's been 40 years. The Apple I — an estimated 200 of which were built — was priced at \$666.66, not because of any devilish influence but because Wozniak thought \$777.77 was too much. For the price, you got a machine that was powered by a MOS Technology 6502 processor skipping along at 1.023 MHz. It came with 4 KB of RAM (upgradable to 65 KB), a monochrome monitor capable of displaying 40 x 24 characters of text, and built-in BASIC.

For storage, there was an interface for a cassette recorder. Keyboard not included. If you are one of the estimated 34 people who still have a functioning Apple I, be aware that one sold last year at auction for \$200,000. **NV**



■ The Apple I, circa 1986.

In this column, Tim answers questions about all aspects of electronics, including computer hardware, software, circuits, electronic theory, troubleshooting, and anything else of interest to the hobbyist. Feel free to participate with your questions, comments, or suggestions. **Send all questions and comments to: Q&A@nutsvolts.com.**

- **Commodore 64 Problem**
- **Reliability Tutorial**
- **Mailbag**

Commodore 64 Video Problems

Q The recent interest in the electronics community in reviving old computers caused me to pull out my beloved Commodore 64 and the associated ADC and voice synthesis boards I built for it.

I was looking forward to seeing the C64 Basic screen on a decent monitor, and used the NTSC composite out in the 64 into a composite input on the HD TV I use for my Raspberry Pi monitor. Much to my surprise, I received the message "Invalid Signal." I changed output channels on both the 64 and the TV, but the message remained. I then took the 64 to my living room HD TV and tried that input with the same result.

Has the NTSC composite signal changed since the 1980s and, if so, how can I build something to allow me to use the HD TV monitor with my 64?

Daryl Isbell
via email

A The *Nuts & Volts* operations manager is actually kind of a Commodore nut, so I'm going to let him answer this question for me.

Tim

As I read your problem description, it made me wonder about how you are setting up your Commodore 64. The C64 does indeed have an NTSC output, but that

signal does not appear on a "standard" jack on the back of the C64 itself. The Video Out jack is a circular eight-pin version and requires a special eight-pin to three RCA style connectors and should look something like the ones shown in **Figure 1**.

This cable is typically used to connect to an "S-Video" Chroma/Luma connection such as that on the back of a Commodore 1702 monitor shown in **Figure 2**.

On the other hand, the RCA jack on the back of the Commodore 64 puts out radio frequency (RF) signals on channel 3 or 4, and is connected like the one shown in **Figure 3**.

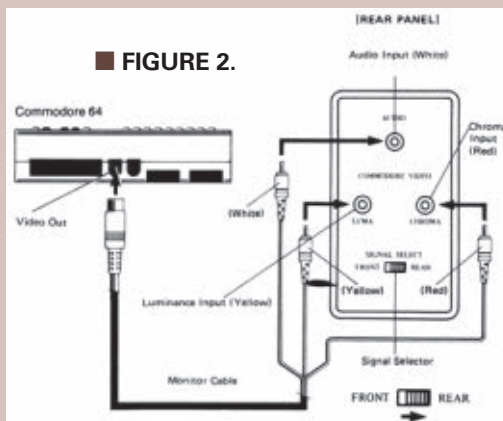
As 300 ohm connectors are rarely found on the back of today's TV sets, a better way to connect your old C64 to new TV sets is to use a special cable to connect to the SHVS (or S-Video) connector on your TV. Instructions for building the cable are available at [ilesj's blog](#) (see Q&A **SIDELINES**). Or, you can buy one off of eBay as they seem to be a readily available item (See **SIDELINES**).

Of course, this is assuming that you connected the RCA jack on the back of the C64 to the NTSC composite (yellow) Video In connector on your HDTV. If that isn't the case and you were using the eight-pin DIN connector and proper video cable, then it might be that the signal from your C64 is slightly out of range due to the age of the internal video generation system. Discussion of this problem can be found on the Lemon64 forum (see



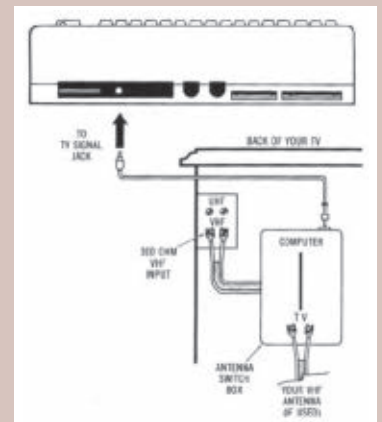
■ FIGURE 1.

Image credit - www.softwareoldies.com



■ FIGURE 2.

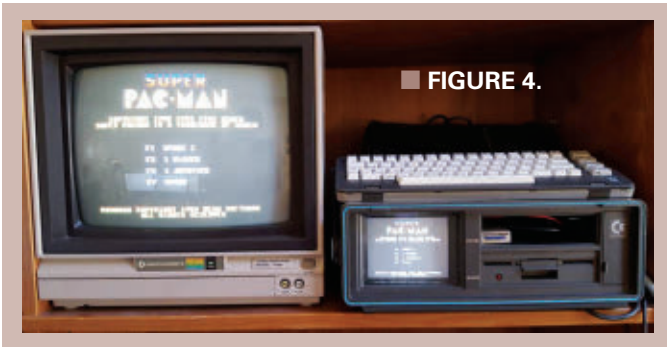
Image credit - www.lemon64.com



■ FIGURE 3.

QUESTIONS and ANSWERS

Post comments on this article at
www.nutsvolts.com/magazine/article/march2016_QA.



SIDELINES).

I hope this gets things moving for you. For the record, **Figure 4** shows what is on the shelf behind me at this very minute.

That's a completely refurbished Commodore SX-64 with an SDIDE and an original Commodore 1702 monitor connected with a Chroma/Luma cable for best picture quality! I hope I have helped answer your question. Please let me know if you get your old Commodore to put out some video.

Vern Graner
Vice President of Operations
T & L Publications, Inc.

Reliability Tutorial

Q I have built and used many electronic devices over the years, which have had varying useful lifetimes. I have always wondered how to determine the lifetime of such devices. Could you explain how to predict the reliability of an electronic device?

Jim Martin
Valparaiso, IN

A Reliability is a quality which is highly sought after by most designers, builders, and users of electronic and mechanical devices. Reliability is essentially the ability of a device to perform a specified function for a certain amount of time with a particular degree of accuracy. Reliability is usually rated in the ability to resist failure. Failure is anything that prevents the device from performing its function over the design life with the desired accuracy. Any discussion of the reliability of a device must first address what is meant by failure and any possible failure modes.

For example, a transistor radio fails anytime you cannot hear a particular station which you have been able to receive in the past. Failure modes could be due to the failure of discrete components such as resistors, capacitors, inductors, diodes, transistors, wiring, speakers, circuit boards, etc. Each component can then have several failure

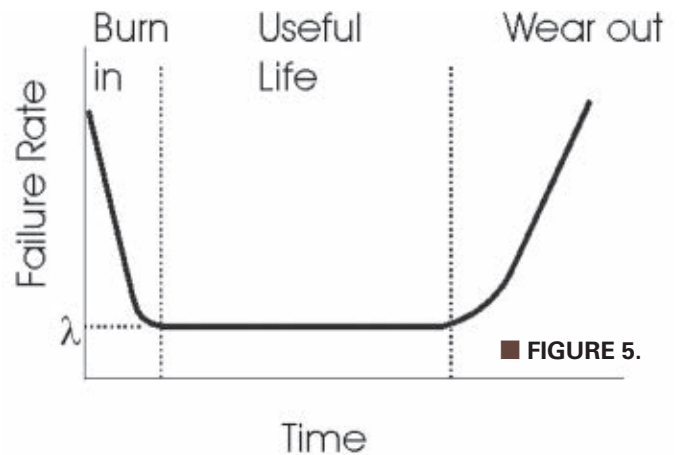
modes, such as a resistor can fail open (burned out; also, I have found a carbon composition resistor invisibly cracked but it was open and the device would not operate), shorted (wire wound turns short together), or there is tolerance drift with time.

Reliability requirements can be different for similar circuits depending on the end use of the circuit. For example, if a device is designed as a one-time use device for entertainment, the cost of improving the circuit's reliability may be too much to allow the device to be sold competitively for a reasonable profit. At the other end of the spectrum, if the device is embedded into a spacecraft system on a deep space probe where there is no chance of repair and the costs of the probe are in the millions of dollars, the reliability will be maximized to the highest level achievable.

The cost of improving can increase exponentially after we reach a specific level using available technology. Devices that we build on our workbenches are most likely not candidates for high reliability. We can improve the reliability of our circuits by using high quality components (e.g., mil spec) and careful assembly techniques, but these will increase the cost of our projects.

Reliability of components is not related to their specific tolerances. Tolerance is a measure of the accuracy to which the component is manufactured, where reliability is a measure of how well the component performs its function under a given set of conditions for a specified period of time.

Reliability of components can be visualized as a failure rate curve as shown in **Figure 5**, which is often called the "bath tub curve." When a large group of components is put into service, there are many components that fail early in the life of the device as seen in the "burn-in period" of the failure rate curve. For critical systems, some electronics



Typical Failure Rate Curve

λ = failure rate during the useful life

Image Credit - www.ece.cmu.edu

manufacturers accomplish an accelerated component burn-in by placing the components into an oven and operating the parts.

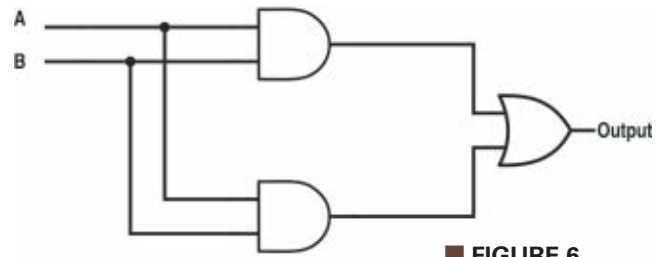
I once visited a plant manufacturing computer control systems, and they put ICs in operating/measuring circuits into a heated oven and operated them for a period of time with the failed parts being rejected.

Check out Paul Verhage's Near Space columns in *Nuts & Volts* to see some of the methods of burn-in testing which are available to and doable by hobbyists without breaking the piggy banks. Due to the loss of the least reliable components, the components that pass the burn-in are much more expensive.

The longest portion of the failure rate curve is the "useful life," which is the time during which the component operates as designed in the device into which it is inserted. The useful life is a statistical value determined by measuring hundreds of components until they fail. The useful life is also called the service life, which is often expressed as mean time before failure (MTBF).

In the real world, reliability was well described by the Yankees great, Yogi Berra: "It ain't over 'til it's over." Or, in electronics terms, no matter how reliable thousands of components have been, the one you have could fail at any time.

The "wear out" period of the failure rate curve for most electronic devices is flat, and the curve ends



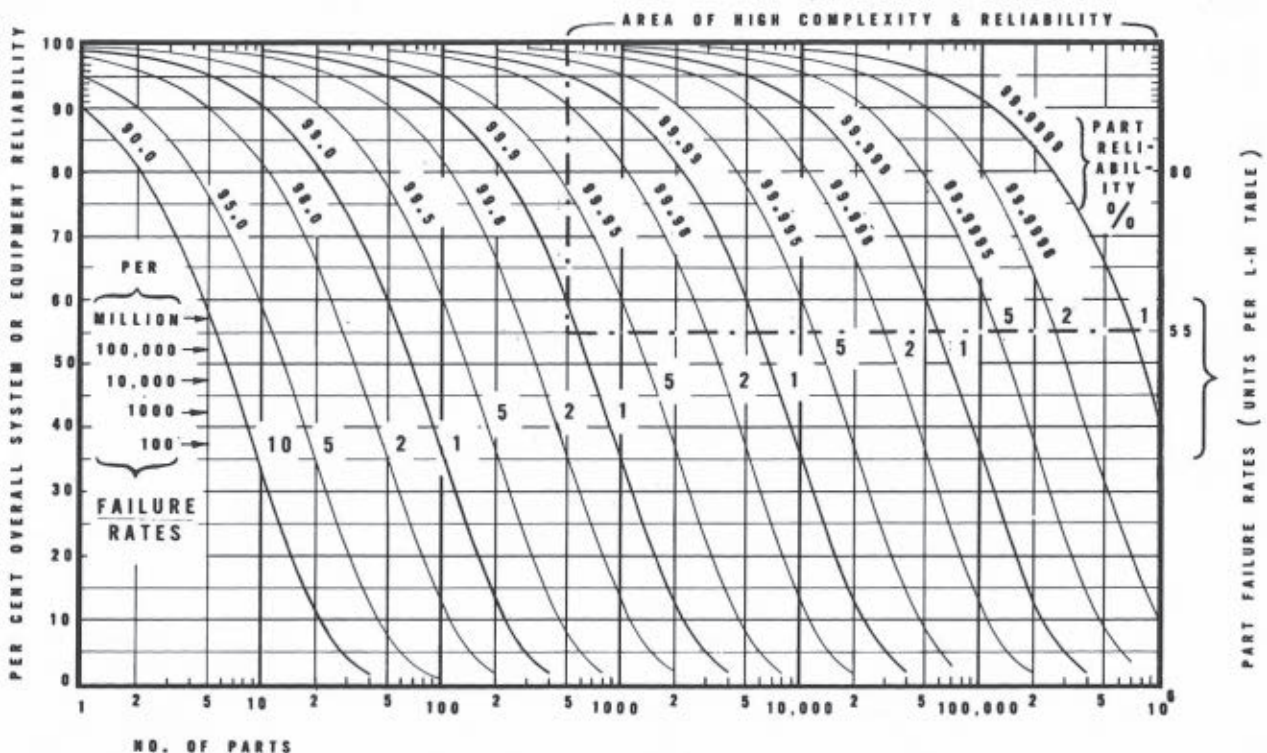
■ FIGURE 6.

Redundant AND Gates

when the component does. Mechanical devices and components with heated filaments (vacuum tubes and incandescent lamps) will experience the wear out shown in the curve.

Reliability of electronic devices is related to the "stress" placed on the component. If we use a 1/2 watt resistor in a circuit in which the resistor only has to dissipate 0.1 watts, the resistor will last much longer than the same resistor used in a circuit in which it must dissipate 0.499 watts. Even this approach has its drawbacks, such as the size of the component.

In our example circuit, a 1/4 watt resistor may fit, but the 1/2 watt one may be too large. Using capacitors rated at higher working voltages can increase their reliability, and operating transistors at lower temperatures will increase their reliability. Protecting a circuit from overvoltage or overcurrent will usually increase its life time. However,



■ FIGURE 7.

Applications Of Reliability Technology System Reliability vs. Parts And Part Reliability

anyone who has witnessed a \$2,500 circuit board burning up to protect its 25 cent fuse knows this does not always work. Surge protection, electrostatic discharge protection, and overcurrent are good ways to improve the reliability of critical circuits.

Redundancy is another way to improve the reliability of an electronic device. Redundancy means that we provide more than one path, component, or device to perform the same function. On the high end of redundancy, the Space Shuttles used five General-Purpose Computers (GPCs) which all ran the same program and periodically checked each other for accuracy. The Shuttles could operate safely on three GPCs, so the other two were extra baggage until a GPC failed. Then, the “spare” GPC picked up the slack.

Figure 6 shows a simple example of redundancy in which two AND gates are processing the same signals with their outputs fed to an OR gate. If one AND gate fails, the other still passes the correct signal. If the AND gate fails with a “hot bit” output, the redundancy does not work. So, you would need a means of detecting the failed AND gate and disabling it. I will leave that to the reader to figure out and submit ideas to Q&A.

The reliability of a circuit can also be affected by entities external to the circuit such as Electromagnetic Interference (EMI; which can be reduced by proper filtering), shielding and grounding, humidity, temperature, shock/vibration, dust/dirt, and corrosion.

A few calculation methods can help us see how different factors affect the reliability of an electronic device, such as the method found in MIL-HDBK-217: Failure rate = $p_{ib} * p_{iT} * p_{iA} * p_{iR} * p_{iS} * p_{iC} * p_{iQ} * p_{iE}$ Failures/million hours where: p_{iT} = Temperature factor; p_{iA} = Application factor (linear, switching, etc.); p_{iR} = Power rating factor; p_{iS} = Electrical (voltage) Stress factor; p_{iC} = Contact construction factor; p_{iQ} = Quality factor; and p_{iE} = Operating environment factor (taken from [www](http://www.reliabilityeducation.com/intro_mil217.html).

[reliabilityeducation.com/intro_mil217.html](http://www.reliabilityeducation.com/intro_mil217.html)) with the p_{iS} representing statistically obtained factors using hundreds (if not thousands) of components.

The cost and effort to obtain these factors is not justifiable for hobbyists, but the equation gives you an idea of how to go about improving the reliability of your projects.

As the number of components in a device or circuit increases, the reliability decreases exponentially with time according to the equation: $R = e^{-nt/m}$ where R is the reliability (probability that a system will not fail over time period t) of a system of many components; n is the number of independent components in the system; t is the time of operation in a given environment before failure; m is the average mean time before failure of the components; and e is the base of the Naperian logarithms 2.71828 (from “Applications of Reliability Technology,” Proceedings of the Fourth Congress of the International Council of the Aeronautical Sciences, Paris, 1964, Spartan, pp 778 - 780).

Even though this equation appeared in an aeronautical publication, it is valid for electronics systems. A graph of the System Reliability versus Parts and Parts Reliability is shown in **Figure 7**, which was scanned from “Applications of Reliability Technology,” p 779.

You can see from this graph that the reliability of a system made up of many components decreases both with the increasing number of parts and the decreasing parts reliability. Therefore, we should try to use the highest reliability parts practical and minimize the parts counts where possible.

I have just grazed the upper crust of the science of reliability, but this can be a starting point for further study. As electronics designers and builders, we can improve the reliability of our devices by using high quality components, limiting the environmental stresses (temperature, humidity, vibration, etc.), and avoiding misapplying our designs. **NV**

MAILBAG

Re: Music Editing

At 74 years old, I still love to read N&V and especially your column. I’m a little rusty at this stuff, but I have two points about your correction on page 78, Nov 2015.

(1) Your fix seems to result in different input impedance for the two channels. L channel looks like 10K to me, while the right channel looks like 60K, and it attenuates the signal into the op-amp by 16 percent. I am not sure if this is a significant problem or not.

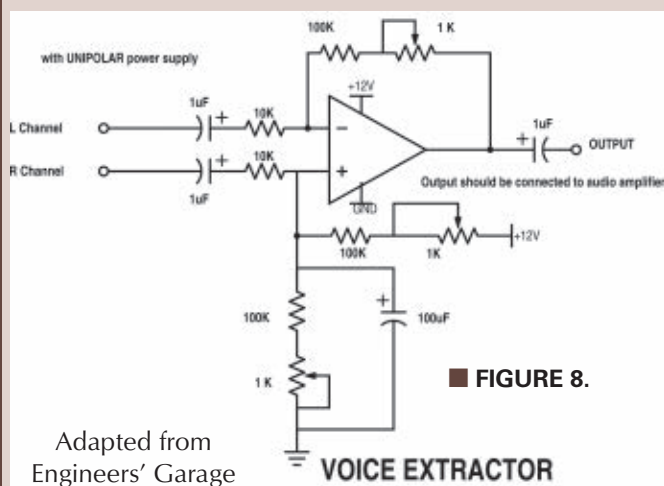
(2) It seems to me that the Engineer’s Garage circuit was designed for a balanced differential input that doesn’t need a common ground between the source and destination circuits (I think). Without this common

ground, the 100 μ F cap would not short the signal to ground.

Again, it’s possible I am way off-base since I haven’t been in the electronics game for a while.

Frank C
via email

*After much analysis, I believe the original circuit shown in **Figure 8** is the most useful one. The 100 μ F capacitor across the 100K resistor bypasses all AC to ground, so the resistor does not affect the AC gain of the amplifier. (Audio signals will flow to the power supply ground even if it is not common to the inputs because somewhere in the circuit there probably is a common*



connection; without the bypass capacitor, the 100K voltage divider resistor will drop voltage, and this voltage will become an extraneous input to the non-inverting terminal.) For the DC, however, the 100K resistor provides a 6 VDC for the output level, so you can use both phases of the input audio signal.

The 100K resistor connected between the +12V power supply line and the voltage divider resistor is the feedback resistor for the R channel. With the same input and feedback resistances for R and L channels, the signals should be balanced and the input impedances equal. There must be an audio common somewhere between the input and the output in order to make a complete circuit for the audio signals in the input channels and the output, but these do not have to be common.

I have added trimmer potentiometers to adjust for resistor mismatch in the feedback circuits and DC level circuit. The "correction" to the circuit proposed in the November Reader Feedback actually rendered the R channel asymmetric with the L channel, so the voice would not be suppressed.

Thank you for your fresh insight into this circuit.

Re: Q&A Tips – Batteries (May 2015)

According to the Huffington Post, freezing alkaline batteries extends their life by only five percent.

David Hadaway
via email

To freeze or not to freeze, that is the question. The March 9, 2010 Huffington Post guest post "Freezer Freaks: 5 Weird Things I Freeze To Save Money," by Jeff Yeager from The Daily Green's says "... for alkaline batteries, freezing extends their shelf life by only about 5%" (www.huffingtonpost.com/2010/03/09/freezer-freaks-5-weird-th_n_491394.html).

I did not see any documentation with the HuffPo post, but from personal experience I have been able to

keep batteries for many years in the freezer and they still performed as well as a new battery. This is a matter of chemistry (the Arrhenius Equation) where the higher the temperature, the more active the electrolytes in the battery are at corroding the metallic elements (electrodes and casing) in the battery. When the electrodes are dissolved, they cannot "make" electricity. When the casing is breached, the electrolyte leaks out, thus reducing the battery's capacity and ruining the device in which the battery is installed.

Remember to warm the batteries to room temperature using the ambient air before using them to insure proper electrical capacity (low temps = low capacity).

Re: Breaker Tripped/Sump Pump Alarm #1

Just perusing the December 2015 column, my eye was caught by Figure 11. I hope you don't have any circuits in your home wired this way! That circuit breaker will always be tripped — tied from hot to neutral — and the alarm will sound constantly. I'd go with either an LED (or optoisolator, if hooking up an audible alarm) in series with a diode and appropriate resistor across the breaker or a neon bulb with appropriate resistor. These will be in series with the load.

Another solution is to have the monitored breaker also power a relay, and tie the alarm via an NC contact to another circuit not prone to tripping.

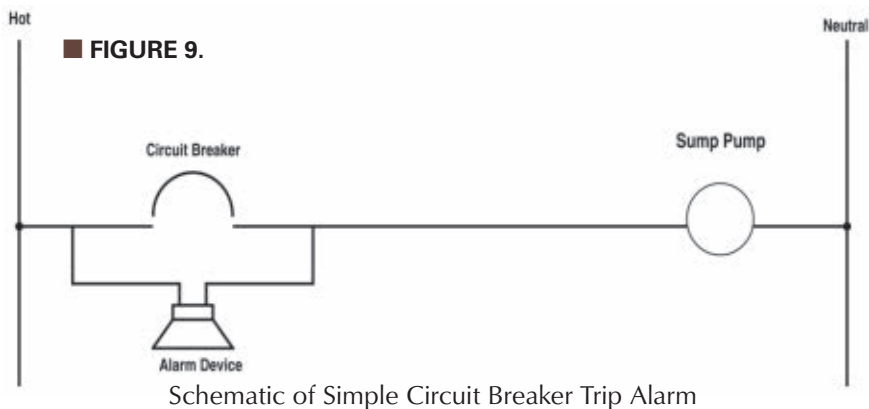
Alan Rauchwerger
via email

You are correct the circuit shown is a dead short, but it was not intended to represent an actual circuit (I left off the circuit's load to simplify the drawing). In the December Q&A, Figure 11 (modified circuit shown here in **Figure 9** with a load inserted) was meant to be an illustration of connecting an alarm device across a circuit breaker, but there should also be a load (sump pump) in the circuit between the circuit breaker and the neutral line.

The device wired to the circuit has sufficient impedance/resistance to prevent the short-circuit. The alarm circuit wired around the circuit breaker would draw current only when the breaker trips because there would be full line voltage across the alarm circuit, whereas when the breaker is not tripped, there is zero voltage across the breaker. A relay, LED/diode, or neon lamp could be used to implement the alarm circuit. I would insert a current-limiting resistor in series with the LED/diode or neon lamp to protect the device, but high enough so as not to shunt too much current from the load.

Re: Breaker Tripped/Sump Pump Alarm #2

I suppose you have already had lots of reaction



“load” for this to work. A wall wart with the appropriate noisemaker across the breaker would be sufficient.

Norm Johnson

*Norm, you are quite correct on both points. There should have been a load added to the neutral line and I have been thoroughly spanked for this omission. **Figure 9** shows the correct circuit rather than my attempt to simplify the alarm circuit attachment. The wall wart with an aural alarm is another good idea. Our readers have a*

to the circuit in Figure 11 of the Breaker Tripped/Sump Pump Alarm. Of course, the “neutral” should actually be

wealth of information and ideas which I love to share in this column.

Q&A SIDELINES

ilesj's blog

<https://ilesj.wordpress.com/2012/11/17/c64-s-video-cable-the-easy-way-and-with-no-scart>

Lemon64 Forum

www.lemon64.com/forum/viewtopic.php?t=27985&sid=167fec3ca9b0f992a036cf030620b9e2

eBay C64 TV/Monitor Cables

www.ebay.com/itm/6-Commodore-Video-TV-Monitor-Cable-S-Video-Audio-C-64-C-128-/231720511410?hash=item35f39e67b2:g:rYAAAOxy4dNS9o1b



Your Single Source of Electronic Equipment, Tools, and Components

01DM820B

Digital Multimeter
Super Economy

DMM - Volts, current, Resistance, Transistor and Diode Test. Includes Leads



\$7.50

01DMMY64

Digital Multimeter
32 Ranges

Including temperature, capacitance, frequency and diode/transistor testing



\$29.95

01DMM9803R

Bench DMM
True RMS

Auto and manual range, RS232C standard interface. Runs on DC or AC



\$139.95

Sweep Function Generator

3MHz, 6 Waveform Functions, Int/Ext Counter, lin/log sweep

Model FG-30 (no display) **\$135**

Model FG-32 (5 digit display) **\$195**



01DS1102E

100MHz Rigol Oscilloscope

Features: 1 million points of deep memory, FFT, record and replay, roll mode, alternate trigger mode, and adjustable trigger sensitivity



\$399

01SDS1052DL

Siglent 50MHz DSO

500MSa/s Sampling
32K Memory
6 Digit Frequency Counter
USB Interface
7" Color TFT-LCD Screen



\$279

DC Power Supplies

Model HY3003

Variable Output, 0-30VDC, 0-3Amp.

Model HY3003-3

Two 0-30VDC, 0-3Amp Variable Outputs plus 5V 3A fixed.



\$95

\$195

Tronix Labs

Cool experiments with no soldering. Includes all Components, Breadboard, Battery, and Manual



Tronix1...Fundamentals

\$28.25

Tronix2.....Digital

\$44.95

Tronix3.....Amps

\$39.95

0603ZD98

48W Soldering Station

Temp. Controlled 310°F-840°F



\$15.25

1 (800) 972-2225

| <http://www.elexp.com>

| contact@elexp.com

More Hot & Sticky

Last time we were together, I shared my summertime adventure to Alaska which included catching salmon and halibut in the waters off of Seward, and dining on that fish with fresh vegetables from my brother-in-law's greenhouse. You don't get closer to the earth than that! In my desire to design and build a greenhouse controller for Lew, I wrote some code for the SHT11 humidity/temperature sensor. It's a nice sensor, but it's expensive. On the other end of the price/performance scale is the DHT11. It doesn't have the range or accuracy of its cousin, but it comes with a user-friendly price-tag: about \$5 from US vendors, and cheaper than that from a host of eBay vendors. The DHT11 (and slightly more accurate DHT22) uses a one-wire protocol, so we'll write a driver for that and have another tool in the box for humidity and temperature measurement.

The DHT11 sensor is bigger than the SHT11 module, yet still fairly small: 12 mm (~0.5") wide by 15.5 mm (~0.6") tall by 5.5 mm (~0.2") thick, and has four breadboard-friendly pins spaced on 2.54 mm (0.1") centers. Only three of the four pins are used as shown in **Figure 1**. If you explore eBay for a few minutes, you'll find dozens of pre-built modules like the one shown in **Figure**

2, which is what I used for my experiments. It includes the DHT11, a bypass cap, and a pull-up on the data line.

Connecting to the DHT11/22 is very easy: We need power and a single free I/O pin. From the host side, the I/O pin is used to trigger the sensor. From the sensor side, this pin is used to respond to the host through a specialized serial protocol. As both sides can control the data line, it is configured as open-collector/open-drain; that is, a pull-up is required to take the line high. When the host wants a new reading, it will pull the data line low. It then releases the pin by changing to input mode to receive a data stream that contains the humidity and temperature values.

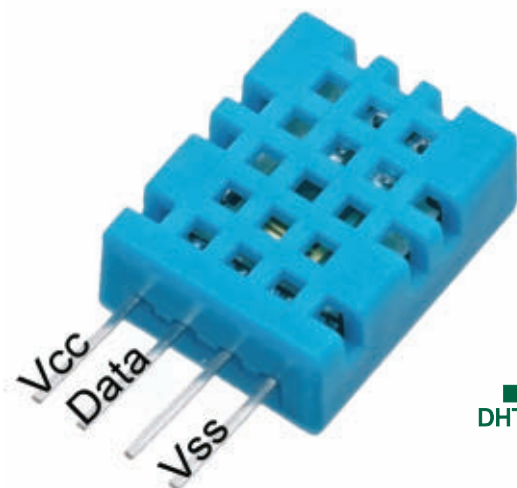
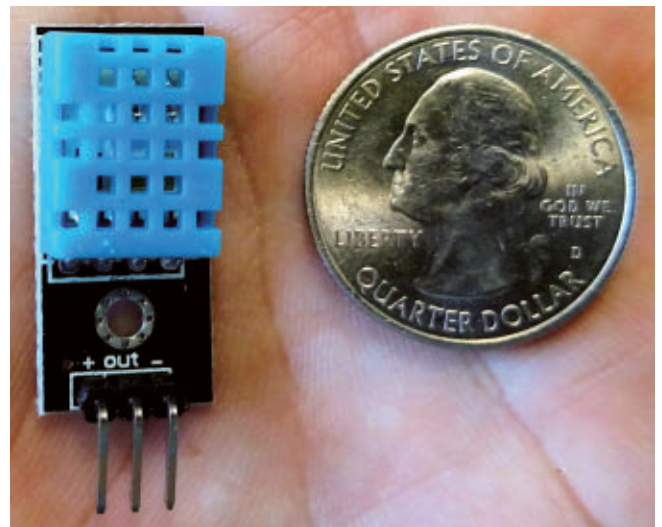
Figure 3 shows the basic connections if you're going the DIY route. Most of the sensors I've seen suggest they run from 3.5 to 5.5 volts, so I use 5V for my setup. The 4.7K pull-up is also connected to 5V. If you're new to the Propeller, this connection to 5V may seem problematic. It's not. The 4.7K pull-up limits the current into the I/O pin to a safe level. **Figure 4** diagrams the transaction elements on the data line. The thin black line denotes control by the pull-up; the blue line is the host pulling the data line low; and the red line is the sensor pulling the data line low:

Trigger: At the beginning of each transaction, the host will pull the data line low for at least 18 ms, then release the data line to the pull-up. This trigger condition brings the sensor out of its sleep state.

Wake: If all is well, the sensor will wake and respond within 20 to 40 microseconds.

Response: The sensor will pull the data line low for 80 microseconds, then release it to the pull-up for another 80 microseconds.

■ **FIGURE 2. DHT11 sensor module.**



■ **FIGURE 1. DHT11 sensor.**

Post comments on this article and find any associated files and/or downloads at www.nutsvolts.com/magazine/article/march2016_SpinZoneTempHumidMeasure.

Data: Sensor values and the checksum are transmitted as a packet of five bytes using bit-width encoding. Every bit is preceded by a 50 microsecond low period. A zero bit has a high period of about 28 microseconds, while a one bit has a high period of 70 microseconds. After the last bit, the sensor releases the data line and returns to its sleep state.

The data bytes in the stream are organized as follows: 0 is humidity (whole units); 1 is humidity (tenths); 2 is temperature (whole units); 3 is temperature (tenths); and 4 is checksum (sum of bytes 0..3). Note that humidity is expressed in percent and temperature in degrees Celsius.

Truth be told, I don't know if I'm going to use this sensor in Lew's project. That being the case, I really didn't want to invest the time in writing an assembly driver. Luckily, with a couple tricks, we can, in fact, write a Spin driver that deals with the DHT11 timing. I've constructed my driver such that it continuously reads the humidity and temperature at a user-specified rate (typically every one or two seconds). A reading counter lets the parent object know a new set of values is ready; if the readings count stops changing, something has gone wrong with the sensor. Have a look at the DHT11 object file (*jm_dht11_ez.spin*):

```
pub start(pin, secs)

    stop

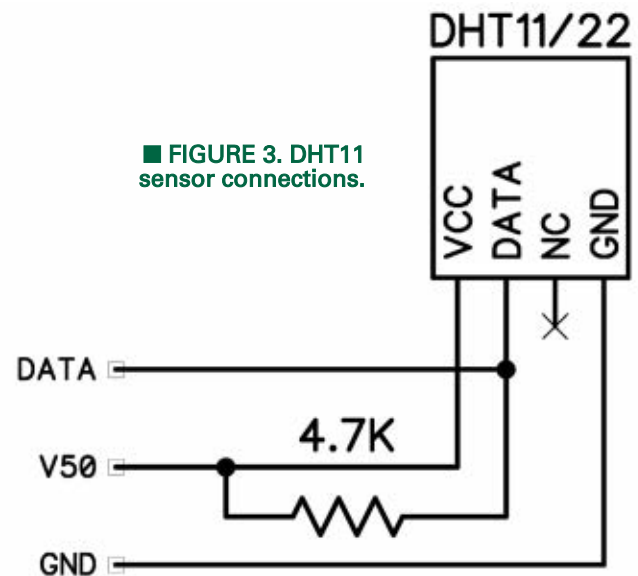
    outa[pin] := 0
    dira[pin] := 0

    cog := cognew(scan_dht11(pin, secs),
    @stack) + 1

    return cog
```

The **start()** method for the object requires the I/O pin be used (this cannot be shared) and a seconds value for the sensor loop. In my case, I use one second, though the DHT22 — which has slightly better specs — should be set to two seconds. Per normal convention, the **start()** method will stop the driver if it's already running. The next step is to clear the I/O pin in the parent cog. Remember that Propeller pins are OR'd together on the output; this means that if one cog makes a pin high, that pin will be high all the time, regardless of what the other cogs do. This would interfere with triggering and reading from the sensor.

With the pin clear, we can



■ FIGURE 3. DHT11 sensor connections.

launch the cog which — in this case — is a method called **scan_dht11()** that runs an infinite loop. Spin cogs require a bit of stack space for overhead. The variable called *stack* in the declaration is an array of 32 longs.

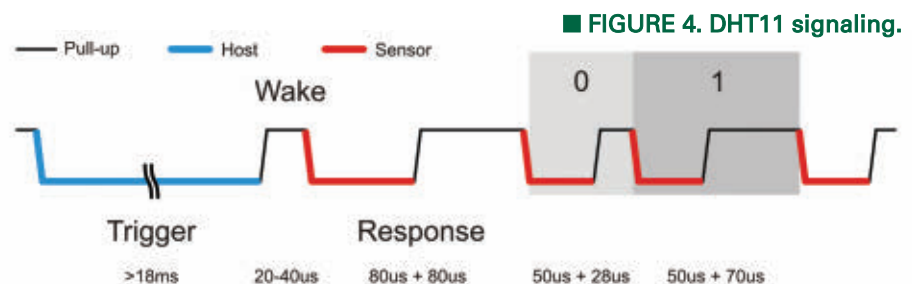
Stack space allocation is always tricky business. I tend to start with the number of parameters plus the number of local variables used in the method, add 16, then round up to the next increment of 16 — this is all contingent on no calls to other methods (as is the case with this object):

```
pri scan_dht11(pin, secs) {
    } | mask, us1, bit1, t, {
    } idx, nbit, raw, cs

    mask := 1 < pin
    secs := (1 #> secs <# 10) * clkfreq
    us1 := clkfreq / 1_000_000
    bit1 := us1 * 35

    ctra := (%01000 << 26) | pin
    frqa := 1
```

My habit is to declare methods intended for their own cog as **PRIVATE** — this reminds me that I should not call



■ FIGURE 4. DHT11 signaling.

them like a standard method; this is particularly true with a method that is intended for its own cog in a child object. To set up for the scan loop, we convert the pin to a mask (for **waitpxx**), limit seconds to 10 and then convert to clock ticks (for **waitcnt**), calculate the number of ticks in a microsecond, then calculate the number of ticks in 35 μs which is going to be our threshold for a “1” bit (a “0” bit is about 28 μs).

We’re going to take advantage of the Propeller’s counter to measure the width of the data bit. For high pulse measurement, we use POS detector (%01000) mode. In this mode, the counter will add the value of **frqa** to **phsa** when the connected pin is high. That’s it for the setup. Now we can drop into the scan and report loop:

```
t := cnt
repeat
  dira[pin] := 1
  waitcnt(t += (us1 * 20_000))
  dira[pin] := 0
  longfill(@raw, 0, 2)

  waitpne(mask, mask, 0)
  waitpeq(mask, mask, 0)
  waitpne(mask, mask, 0)
```

Just before entering the loop, we take a snapshot of the **cnt** register — this provides the initial sync point for loop timing (typically one or two seconds). The first instruction in the loop writes one to the **dira** bit of the pin, causing it to go low. You may wonder why we didn’t write zero to the associated **outa** bit; we don’t have to — when a cog is started, all **dira** and **outa** bits are cleared.

A 20 ms delay is created with **waitcnt** before the pin is released back to the pull-up by writing zero to the **dira** bit. While waiting for the wake and response, the variables **raw** (readings) and **cs** (checksum) are cleared. There’s another subtle trick here. Note that we’re using the address (@) of **raw** for **longfill**, and that the instruction will clear two longs. Spin assigns memory locations for the same variable type in the order they appear in the declarations; in this case, **cs** follows **raw** and gets cleared by the **longfill** instruction.

If all is well with the sensor, it should wake and generate a low-going response pulse of 80 μs , followed by a release to the pull-up of 80 μs . After this, it will output 40 bits (five bytes) of data. We don’t actually need to measure the response pulses; we just have to look for activity. The three **waitpxx** instructions let us know everything is okay. The first, **waitpne**, is looking for the sensor to pull the data line low. The next, **waitpeq**, is waiting for the sensor to release the data line. Finally, another **waitpne** indicates the start of the data stream:

```
repeat idx from 0 to 4
```

```
repeat nbit from 7 to 0
  phsa := 0
  waitpeq(mask, mask, 0)
  waitpne(mask, mask, 0)
  if (phsa ==> bit1)
    raw.byte[idx] |= 1< nbit
```

Receiving five bytes is done using two nested loops: one for the bytes (0..4); the other for the bits (7..0 for MSBFIRST mode) in each byte. Remember that the start of each bit is a 50 μs low period. At this point, it is safe to clear **phsa**, and then use **waitpeq** and **waitpne** to hold for the pulse. After the pulse, **phsa** is compared with the one-bit timing threshold and — if met — that bit in the current byte is set to one using a mask.

Before we continue, let me point out a variation in the RAM addressing trick I mentioned earlier. We have two variables, **raw** and **cs**, declared as longs (all local variables are longs), and we want to use them as an array of bytes. No problem! The **.byte[n]** modifier of a variable allows us to do this. “But there’s only four bytes in a long!” you remind me (because the loop wants to receive five bytes). Yes, that’s correct, but at the core of things, a variable name is simply an expression of a RAM address; we can go from there as we please. What this means is that **raw.byte[4]** is the same as **cs.byte[0]**. By taking advantage of this nature, we’re able to simplify the receive loop. Note that we can even have negative indexes in the modifier; **cs.byte[-4]** is the same as **raw.byte[0]**. That we can do something doesn’t mean we *should* — be careful with negative indexes.

With the packet received, the next step is to validate it using the checksum. Again, this is just the sum of the four bytes transmitted for humidity and temperature:

```
repeat idx from 0 to 3
  cs -= raw.byte[idx]
```

This loop subtracts each of the data bytes in the packet from the checksum byte; if there is no error, **cs** will be zero at the end of the loop. Easy-peasy:

```
if (cs == 0)
  rhtresult := raw
  badchecksum := false
else
  badchecksum := true

++count
waitcnt(t += secs)
```

If the checksum is zero, the packet (in **raw**) is copied to an object-global variable called **rhtresult** and the checksum error flag is cleared. If there was a problem, we don’t update the result, but we do set the error flag.

At the end of the loop, the hub-global variable *count* is incremented, which is a signal to the parent object that a new reading is ready. When *count* is updated, the result and checksum flags are already in place, so it is safe to use them. Okay, then, let's read the humidity and temperature:

```
pub rh
    return rhtresult.byte[0]

pub rh10
    return (rhtresult.byte[0] * 10) {
    } + rhtresult.byte[1]

pub tc
    return rhtresult.byte[2]

pub tc10
    return (rhtresult.byte[2] * 10) {
    } + rhtresult.byte[3]

pub tf
    return tc * 9 / 5 + 32

pub tf10
    return tc10 * 9 / 5 + 32_0
```

As you can see, the access routines are really straightforward. We're simply returning the proper byte for the associated reading. In the case of **rh10()** and **tc10()** with the values in tenths, we will multiply the whole byte by 10, then add in the associated tenths byte.

Using the parent object (*jm_dht11_demo.spin*) is as easy as you'd imagine. We can call the **dht11.readings()** method to see if new values are available. Here's the main loop of the demo program:

```
pub main | last, now, temp, p_str

    setup

    set_pixel_char("@")

    last := 0

    repeat
        repeat
            now := dht11.reading
        until (now <> last)
        last := now
```

```
big_dec3(get_temperature, 0, 0)
xy_str(18, 0, @@TScale[T_MODE])

big_dec3(get_humidity, 0, 9)
xy_str(18, 9, string("%"))
```

The top of the loop calls the **dht11.reading()** method until a change is detected — this indicates a fresh set of humidity and temperature values are available. The temperature is printed on the Parallax Serial Terminal (PST) using **big_dec3()** which wants a value (from **get_temperature()**), and the upper left coordinate of the printed value. PST is fairly simple, but I really like that it has x/y formatting commands which lets us create a nice output through a standard serial connection. I use this all the time for test programs in the products I design. **Figure 5** shows the output from the DHT11 test program.

After I got things working, I noticed that the readings were a little twitchy; this isn't a big surprise considering the low cost of the sensor. What to do? One of my favorite tricks to smooth out noisy readings is to use a rolling average. For the temperature and humidity readings in the demo program, I use a 10 second window. In the VAR section, we have this:

```
var

    long    tavg[10]
    long    tidx

    long    havg[10]
    long    hidx
```

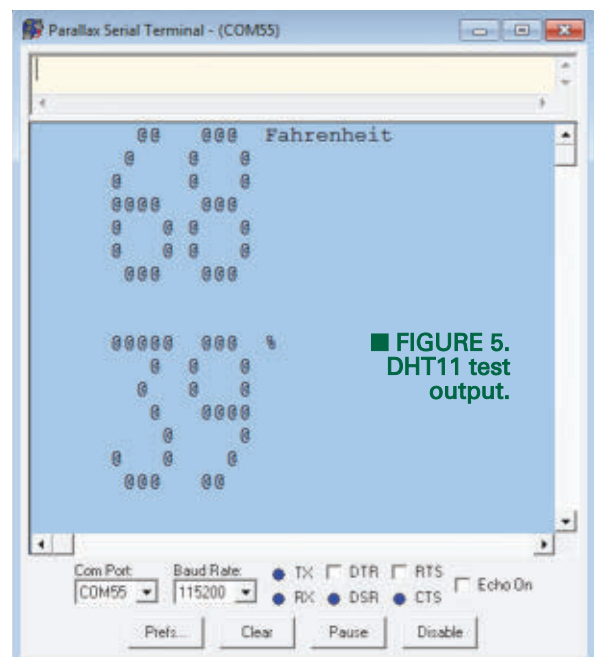


FIGURE 5.
DHT11 test
output.

Item

Propeller Activity Board
DHT11/22 Sensor Module

4.7K Resistor

Source

Parallax #32910
Adafruit DHT11 or DHT22
SparkFun RHT03 (DHT22)

BOM

What we have are arrays that will store the last 10 readings, and an index that tells us where to put the next. When we call **get_temperature()** or **get_humidity()**, we'll take a reading, add it into the array, then average the array and return that value. Here's the code for temp averaging:

```
pub get_temperature | sum, idx

if (T_MODE == TC)
  tavg[tidx] := dht11.tc
else
  tavg[tidx] := dht11.tf

if (++tidx == 10)
  tidx := 0

sum := 0
repeat idx from 0 to 9
  sum += tavg[idx]

return sum / 10
```

The routine uses the program constant called **T_MODE** to determine which temperature routine to call:

Celsius or Fahrenheit. Once the reading is added into the array, the index is updated, and then checked for roll-over. When this is the case, the index is reset. The values in the array are then summed, and the average returned to the caller. Using a one-second scan loop and a 10-element averaging array, you'll notice that it takes 10 seconds to reach the normalized reading. We could modify the averaging loop based on the current sensor reading counts, but I find this an unnecessary complication.

Look @@ that Data!

Printing the label for the temperature mode employs an interesting addressing technique in Spin using the @@ operator. You know that @ provides the absolute address of an object in RAM when the program is running. When used in a constant expression, however, it only returns the symbol's offset within the object. The @@ operator allows us to get the absolute address of the symbol that uses @ in a constant declaration. Consider these strings:

dat

Celsius byte "Celsius", 0

The Easiest Way to Design Custom Front Panels & Enclosures

Free
Front Panel
Designer



You design it
to your specifications using
our FREE CAD software,
Front Panel Designer

We machine it
and ship to you a
professionally finished product,
no minimum quantity required

- Cost effective prototypes and production runs with no setup charges
- Powder-coated and anodized finishes in various colors
- Select from aluminum, acrylic or provide your own material
- Standard lead time in 5 days or express manufacturing in 3 or 1 days

**FRONT PANEL
EXPRESS**

FrontPanelExpress.com

We Offer the Lowest Prices on the Best Scopes



Passport-Size PC Scopes \$129+
Great scopes for field use with laptops. Up to 200MHz bandwidth with 1GSa/s, high speed data streaming to 1MSa/s, built-in 1GSa/s AWG/function gen. **PS2200A series**



30MHz Scope \$289
Remarkable 30MHz, 2-ch, 250MS/s sample rate scope. 8-in color TFT-LCD and AutoScale function. Includes FREE carry case and 3 year warranty! **SDS5032E**



50MHz Scope \$399
50MHz, 4-ch scope at 2-ch price! Up to 1GSa/s rate and huge 12Mpts memory! Innovative "UltraVision" technology for real time wfm recording. FREE carry case! **DS1054Z**



60MHz Scope \$349
Best selling 60MHz, 2-ch scope with 500MSa/s rate plus huge 10MSa memory! 8-in color TFT-LCD. Includes FREE carry case and 3 year warranty! **SDS6062V**



200MHz - 1GHz Scopes \$1,500+
2/4 channel, 8 or 12 bit with long memory, powerful debug capabilities (Teledyne LeCroy) **WaveAce 2000 Series**

- Free Technical Support
- Excellent Customer Service

Saelig
unique electronics

```
Fahr    byte    "Fahrenheit", 0
```

We could use an **if-then** structure to select a string based on the program mode, but if our string list goes beyond a few elements, this can get unwieldy. Another approach is to create a table of string addresses:

```
TShow    word    @Celsius, @Fahr
```

When used in a table like this, @ provides the offset from the start of the object, not its absolute address in memory which is what we need to string methods. This is where the @@ operator comes in: It adds the program offset of the object to an address table element so we get the absolute location of the symbol. We can use the table defined above like this:

```
term.str(@@TShow[T_MODE])
```

Again, using @@ is particularly useful with multi-element string tables that have different lengths. Here's another good example.

```
dat
```

```
Sun    byte    "Sunday", 0
Mon    byte    "Monday", 0
Tue    byte    "Tuesday", 0
Wed    byte    "Wednesday", 0
Thu    byte    "Thursday", 0
Fri    byte    "Friday", 0
Sat    byte    "Saturday", 0
```

```
Day    word    @Sun, @Mon, @Tue, @Wed
        word    @Thu, @Fri, @Sat
```

The program keeps track of the day with a simple number (0..6), and can print any day's name using @@:

```
term.str(@@Day[didx])
```

Do have a look at the demo program; the @@ also helps in printing of big digits to the terminal window. If you haven't explored humidity and temperature measurement due to the cost of precision sensors, the DHT11 is an inexpensive gateway to a lot of fun. Who knows, you might end up building your own greenhouse controller.

Until next time, keep spinning and winning with the Propeller! **NV**



AP CIRCUITS
PCB Fabrication Since 1984

As low as...

\$9.95
each!

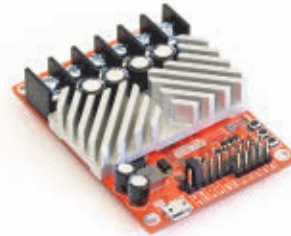
Two Boards
Two Layers
Two Masks
One Legend

Unmasked boards ship next day!

www.apcircuits.com



Motor Control



- 15 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial

- 45 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



- 160 Amps Per Channel
- Dual Channel
- Quadrature Encoders
- DC Brushed Motors
- USB / RC / Serial



www.ionmc.com

NEW PRODUCTS

- HARDWARE
- SOFTWARE
- GADGETS
- TOOLS

NEW VERSION OF WIRELESS DEV PLATFORM

Anaren, Inc.'s Wireless Group announced the upcoming release of version 1.5 of its Atmosphere development platform and is already working to develop the features of what will be version 2.0, including expanded project platform selection and new elements for Wi-Fi. Version 2.0 — a major release — is expected to launch this spring.



The combined local and online software tool set helps manufacturers at every RF design skill level build applications for mobile devices and programs for embedded devices simultaneously. This capability allows developers to quickly design an embedded system and turn a mobile Bluetooth® Smart device (such as an iPad, iPhone, or Android-based device) into a mobile control panel to interact with the embedded system. The software platform works with the AIR for WICED™ Smart Bluetooth modules, developer board, and Anaren's AIR modules. The modules are low cost, surface-mount radio modules that feature Bluetooth Smart and Wi-Fi technology.

The version 1.5 release features better connector highlighting for workflow, the availability of new elements for mobile apps, and Phone gap plug-in support for these new elements, such as Vibrate, Beep, Device Info, Network Info, Alert, Confirm, and Prompt. The upgrade features a substantial Android BLE improvement and other performance enhancements, along with several bug fixes. In addition, a Geolocation element is slated to be included.

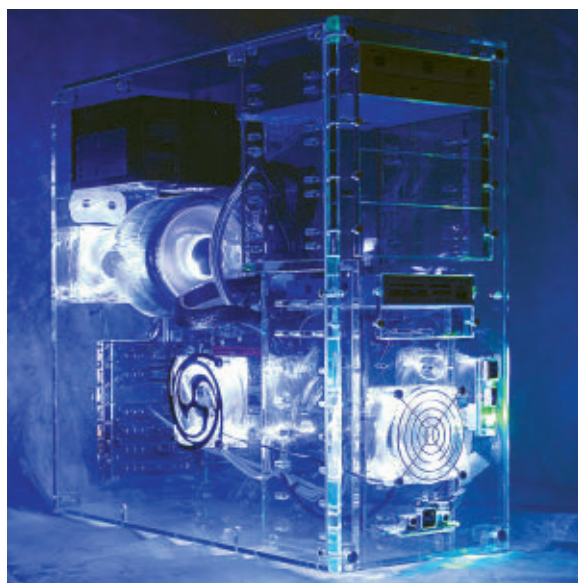
For more information, contact:
Anaren, Inc.
www.anaren.com

FOUR COLORS IN ONE ENERGY EFFICIENT PACK

Super Bright LEDs now offers two surface-mount LED components that have the traditional RGB LEDs with an additional cool or warm white chip for superior color blending. These four-chip Epistar 5050 SMD LEDs are able to produce true white illumination without RGB color bleeding or the need for an additional LED component.

Applications include computer case lighting, case modding, circuit board lights, indicator lights, custom light strip building, light panels, art projects, and monitor or TV backlighting.

Using only 0.2 watts of power, both energy-efficient LED components produce 2,400 millicandelas (seven lumens). The LED components emit a wide 120 degree beam of red, green, blue, and cool white (6,500K) or warm white (3,100K) illumination. Each LED chip color has an individual voltage: Red has a forward voltage of 2.2V, and green, blue, and white have forward voltages of 3.2V.



These LED components are designed to be soldered onto a printed circuit board and can be used in series, but resistors are recommended for overdrive protection.

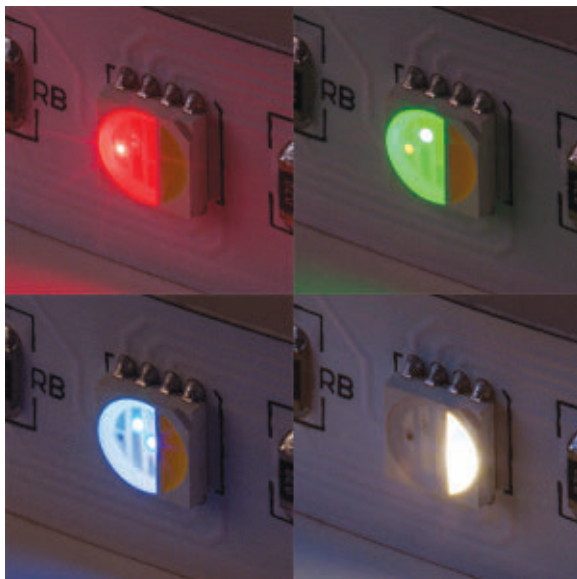
For more information, contact:

Super Bright LEDs

www.superbrightleds.com

If you have a new product that you would like us to run in our New Products section, please email a short description (300-500 words) and a photo of your product to:

newproducts@nutsvolts.com



ENHANCED EFFECTS LIGHT WITH **CANDLE BI-PIN LAMP**

J2 LED Lighting, LLC announces the availability of their EEL1 socket cable, which is designed for mounting into High Density Urethane (HDU) styling and modeling board. The EEL's cable socket fits into a mounting hole from the material's front side. The tapered fit of the socket's body locks it into place, and a front 5/8" diameter flange provides an insertion stop. Lock ribs on the taper provide for anti-rotation of the socket in the material.



Continued on page 65

NATIONAL RF, INC.

TYPE **RNF** HFDF
NOISE LOCATION SYSTEM



LOCATE NOISE IN THE HF SPECTRUM
Visit www.NationalRF.com for this and other Radio Products!
Office: 858-565-1319

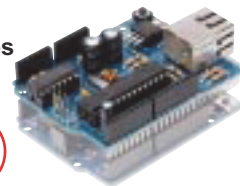


QKITS LTD

sales@qkits.com

Arduino • Raspberry Pi

Power Supplies
MG Chemicals
RFID



\$8.50

flat rate shipping

1 888 GO 4 KITS

Visit us at: www.qkits.com



NKC electronics

MDE8051 Trainer
by Digilent

NKCElectronics.com/MDE8051

Includes the MDE8051 training board, power supply, serial cable

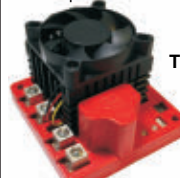
Purchase Orders are accepted from Educational Institutions, US Government and Research Centers



ROBOT POWER Extreme Motor Speed Control!

MegaMoto GT- Mega_Motor Control for Arduino™

- ◆ 6V-40V - 35A / 50A+ peak
- ◆ Single H-bridge or dual half
- ◆ Current and Temp protected
- ◆ Jumper select Enable & PWM
- ◆ Robust power terminals



The VYPER™

- ◆ 8V-42V - 125A / 200A peak!!
- ◆ Single H-bridge
- ◆ RC and Pot control
- ◆ Overload protected
- ◆ For your BIG bots

www.robotpower.com

Phone: 360-515-0691 • sales@robotpower.com

PiBooth — A Raspberry Pi based PHOTO BOOTH



My fiancée thought a photo booth would be a fun addition to our wedding reception. She had a table with some props and a cheap camera for people to use in mind, but it gave me an excuse to start a project and build a fully automated portable photo booth. In classic style, an offhand comment about something turned into a few weekends of solder smoke, coffee, and coding.

Photo booths are a lot of fun; in fact, most of us can probably remember using them with friends or family and getting those small strips of pictures. It turns out that photo booths were conceptualized in the late 1800s and made a public debut at the World's Fair in 1889. Shortly after that, they began to show up on streets around the world before becoming a staple of shopping malls and ID card stations. The earliest photo booths used traditional chemical developer techniques that were messy and hazardous, but luckily for me, digital cameras can now fill that role. I wanted to make a digital photo booth that can store and tweet photos for guests.

The requirements I came up with for my project were simple. I needed an easy-to-use photo booth that could be set up by anyone in just a few minutes. It also needed to be portable since I currently live in Pennsylvania and the wedding was in Missouri. The photo booth should take a series of photos so people can try different props and poses. A live preview of the photo was also essential so that nobody ended up headless in our album.

The particular occasion and venue posed a few additional challenges. The venue had very variable lighting and we didn't want dark outlines as photos, so the photo booth would have to provide its own lighting. It was unclear if Wi-Fi at the venue would be adequate, so photos would have to be stored locally with the option to post to social media, as well.

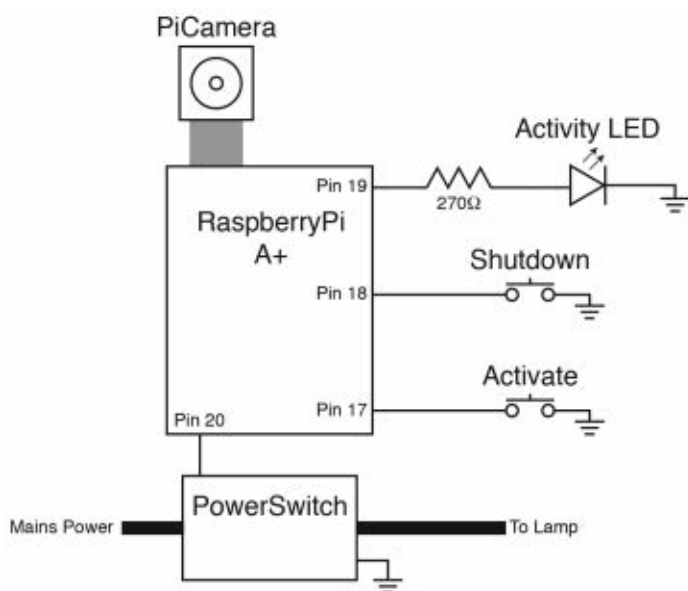


■ **FIGURE 1.** This completed photo booth keeps that old-time feel, while making digital photos available to guests just seconds after they are taken.

To fit with the wedding and my general taste, it needed to have a vintage look and feel. The final two constraints were purely practical: It needed to be inexpensive and relatively quick to develop. Anyone who has planned a wedding knows the cost and time that goes into it, and this project could only excuse so much centerpiece duty. The final result (**Figure 1**) met all of these requirements and was surprisingly easy to build!

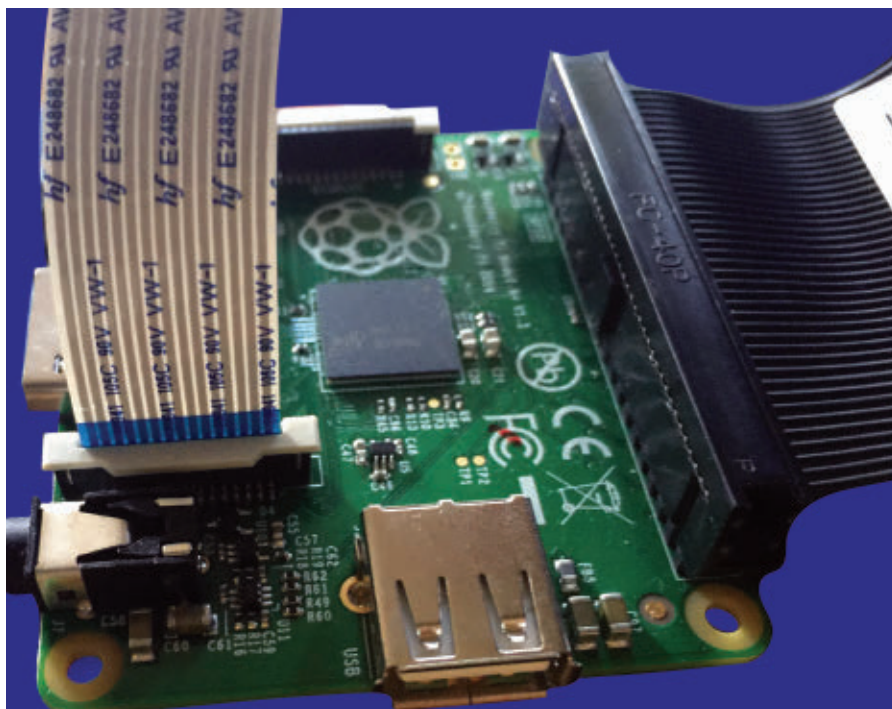
Hardware Setup

When starting a project like this, it is always a good idea to see what's in the parts bin already. In my case, I pulled out a Raspberry Pi A+. While the A+ model isn't good for RAM intensive applications, I didn't plan on loading the graphical user interface (GUI) for the operating system. It is also a small board and very inexpensive. With the large support community behind it, the Raspberry Pi is also very easy to set up and use.



■ **FIGURE 2. Hook-up diagram for the PiBooth. Be sure to keep connections short where possible to reduce the false triggers that the software must fight.**

The first step for me was getting all of the hardware prototyped so that I could start developing the software, and thinking about what kind of enclosure it should all go into. This meant I needed to select a camera and decide on the user interface for the booth.



■ **FIGURE 3. The camera connects to the CSI bus through a flat-flex cable. On the model A+, the bare contacts are towards the HDMI connector. Make sure this isn't backwards!**

The initial solution for a camera was going to be an extra Nikon D40X. This — or any other digital single lens reflex (DSLR) camera — would produce very high quality photos with fantastic automatic exposure settings. The problem was that getting a live preview from these cameras is very difficult on most; impossible on some. Even triggering the photo capture was a bit hacky, as on some models I would need to emulate the infrared (IR) remote signal.

A webcam was a cheaper alternative with many of the desirable qualities, but they can be difficult to interface with. It could be done, but the clock was ticking! The optimal solution for me was the Raspberry Pi camera board. This \$30 camera plugs directly into the Pi with a ribbon cable that breaks out the camera serial interface (CSI) bus. It will take still photos or video, and it has a really nice set of libraries to control it from the Linux shell or through a Python program.

There is no free lunch, though. The camera is only five megapixels and doesn't have automatic exposure settings. However, for the size of prints that guests would make and for social media posting, the resolution was plenty. It also has the bonus of smaller file sizes, which comes in handy when dealing with hundreds of photos.

The photo booth also needed a "big red button" that begged to be pressed to start the photo sequence. This needed to be obvious, intuitive, and make that satisfying tactile click. A less obvious button was needed to shut

down the photo booth. I certainly didn't want to connect a keyboard every time, but this button needed to be less obvious and hard to trigger by accident.

As we all know, every good project has to incorporate some blinking lights, and this one is no different. I wanted a flashing light to replicate that of normal cameras when they are put into time delay mode. It also would help draw people's eyes to look at the camera and not at themselves on the preview screen.

Lighting the photo scene needed to be simple and safe. I considered LED strips, but worried about how to best position them. In the end, I decided on just providing a plug that we could use to switch any lamp we deemed appropriate when setting up. I started building up the transistor/relay circuit to switch the mains power, but found that a product called the "Powerswitch Tail 2" existed. This solved the problem in an opto-isolated and fully enclosed way, and made me feel a lot better about leaving the unit unattended. While my mains wiring would be fine, I didn't want a

Randomly Tweeting Text

Initially, I had the photo booth tweeting the photos with some default text that is stored in the `tweet_text` variable. You can still do that, but if you put a text file in the PiBooth directory that has different tweets on each line and set `tweet_text` to that filename, the photo booth will randomly select a tweet from that file and make your Twitter stream a little more interesting. Below are some examples of the tweets that we used. You can follow my photo booth's adventures by following @Pi_Booth on Twitter.

```
These people really pushed my buttons.  
... and all I got was these photo booth pictures.  
Awww.  
Hey, I just met you, this is crazy.  
I wasn't lucky, I deserved it.  
I had fun once, it was horrible.  
Smile :)  
Hi there!  
OMG that's so cute!  
Collect moments, not things.  
These people ...  
Best selfie ever!  
Frankly my dear, I don't Instagram.
```

curious guest to be able to find AC connections anywhere.

After getting the components I didn't have, I began to breadboard the design. It is really a very quick and simple circuit. An LED with a current-limiting resistor for the flashing timer light, the two leads to the power switch, and two buttons for trigger and shutdown were all we needed. The general-purpose input/output pins (GPIO) on the Pi are very configurable, so I chose to have the lines held high by the Pi and pulled low by the buttons.

A breakout board like the Pi T-Cobbler makes connecting everything to the right pins a lot more simple and easy to change later, but isn't necessary. You can choose whatever GPIO pins you wish, but I used those marked in the hook-up diagram shown in **Figure 2**. That is what the downloadable code (available at the article link) will be set up for.

The camera hooks up to a flat-flex connector on the Pi. Be careful that you connect the camera to the correct connector and in the right direction! On the Pi A+, the bare contacts go towards the HDMI connector and the blue strip is towards the USB connector (**Figure 3**). Gently lift the connector bail and insert the cable, then snap the bail back down. A small screwdriver could help if you have large fingers.

For initial prototyping, I hooked the Pi up to my living room television, but it is important to keep testing it with the screen you will use in the end product to catch any resolution issues early. Use a decent quality HDMI cable or you may run into reliability issues.

It is easiest to work with the Pi — at least initially — if it is connected to a keyboard, mouse, and the Internet. I used a small USB hub to break out the single USB port on the A+. Powered hubs are really helpful if you are running

into erratic behavior. Wi-Fi modules can consume a decent amount of power and could be the root of the problem. Power is the last thing our machine needs. Again, use a high quality USB cable for durability and because they generally have larger gauge wire than very cheap cables. I used an old Apple iPad charger as my power source. Whatever you use should be able to source about 2A at 5 VDC.

Software

There are a lot of steps to get all the parts in place for the photo booth to work flawlessly and automatically. Hang in there, though; it is worth it!

To get started with your Raspberry Pi, you'll need to set up the operating system (Raspbian) on an SD card. The easiest way to do this is with a tool called NOOBS (New Out Of the Box Software). You can buy SD cards with NOOBS already installed, or do it yourself. I chose to do it myself on a large (32 GB) SD card to provide lots of photo storage. You can format the SD card with tools already on your computer, or with SD Formatter from the SD Association (www.sdcard.org/downloads/formatter_4).

Download the offline and network install version of NOOBS from the raspberrypi.org site (http://downloads.raspberrypi.org/NOOBS_latest) and unzip it. Copy the contents onto your SD card and eject it from your computer. Insert the SD card into your Pi and power up.

You'll be greeted with a window that allows you to select the OS you wish to install. Installation can take a while, but eventually will complete and an onscreen configuration guide will take you the rest of the way. Be sure to enable the camera during the prompts! You can now log in and load up Raspbian (see the **sidebar**).

Before continuing, it's a good idea to update the software on your Pi since the NOOBS distribution and packages could be a little dated. Make sure you've set up the network connection and at a terminal prompt, type "sudo apt-get update," then "sudo apt-get upgrade." Again, this took a while for me to complete.

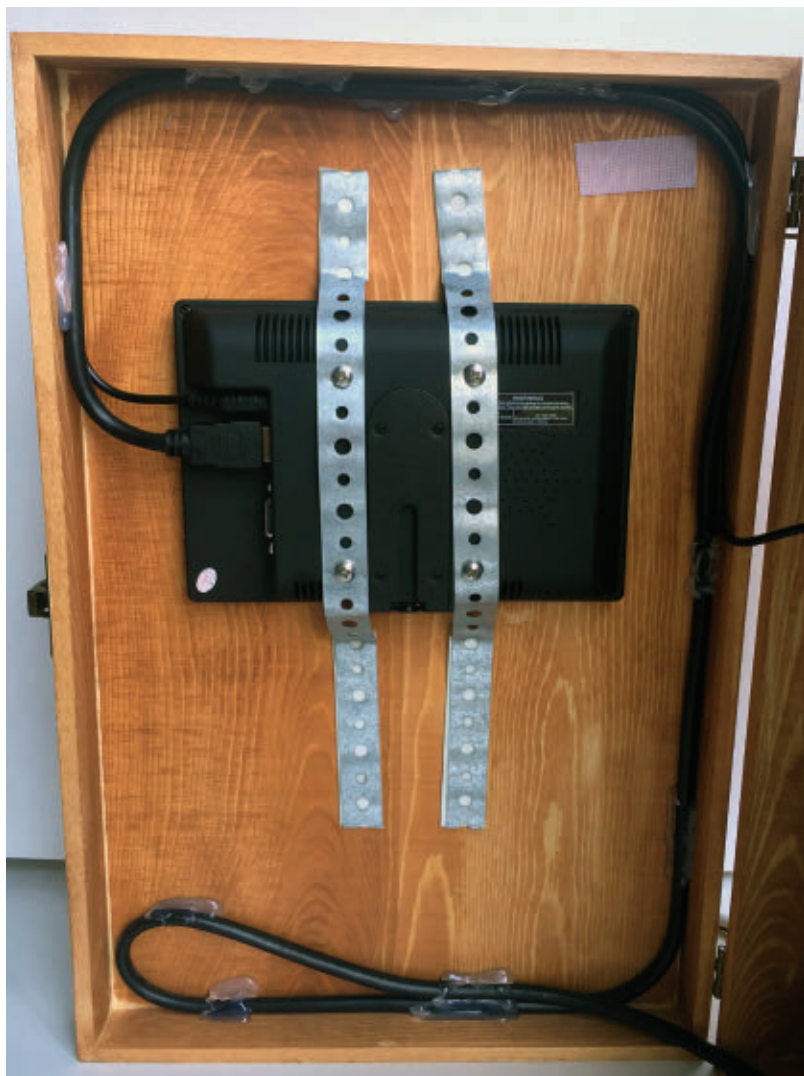
While we have the Pi connected to the Web, we'll

Logging into Your Pi

It's not uncommon to boot up a new system and not know how to log in. The current distribution of Raspbian has the following default login:

```
username: pipassword: raspberry
```

It is a good idea to change the default password — especially if you are going to make your project Internet connected. Just type "passwd" at the terminal prompt and follow the instructions. You can even create a new user, but be sure to modify the setup instructions to use that account instead of the Pi account.



■ **FIGURE 4.** The 7" monitor can be securely mounted with plumber's tape, some metric screws, and heavy duty double-sided tape. Notice the cable routing and Velcro for the TV remote.

the Twitter client, so run "sudo apt-get install python-pip." Now, install the Python Twitter package, Twython: "pip install twython." Finally, navigate to the home directory ("cd ~") and clone the software repository by typing "git clone <https://github.com/jrleeman/PiBooth.git>." With that, you have installed all the software you need.

Now, we need to set up Twitter authentication. If you don't want to use Twitter, you can skip this with no consequence. Sign into your Twitter account and go to <https://dev.twitter.com>. At the bottom of the page, click "Manage Your Apps." Here, you can create a new app.

Enter a name for the photo booth and create the application. In the "Keys and Access Tokens" tab, you'll find the Consumer Keys. Below the Consumer Keys you can generate the Access Token. Copy and paste the Consumer Key, Consumer Secret, Access Token, and Access Token Secret into their respective places in the photobooth.py file you downloaded from GitHub. While there, change the "tweet_photos = False" to True. You can even set up random Tweet captions for your photos (refer to the sidebar).

If you choose to tweet, you'll also need to set up the network connection. I opened the GUI for the operating system by typing "startx" at the command prompt, set up the network, and then rebooted.

If you want to understand how the Python code does everything, you should dig around and play with it. The pygame module is used to do text overlays, show a countdown to the photos, play a shutter sound when the photo is taken, and to handle key press events. I didn't expect to change the contrast and brightness

download the photo booth software and dependencies, as well. We need the Python package manager (Pip) to install

settings often, but didn't want them hard-coded. They are tied to the arrow keys with pygame. The text overlays

ITEM	PRICE	SOURCE
Powerswitch Tail 2	(\$25.95)	www.adafruit.com/products/268
Raspberry Pi Camera Board	(\$29.95)	www.adafruit.com/products/1367
Flex Cable for Raspberry Pi		
Camera — one meter	(\$3.95)	www.adafruit.com/products/2143
Big Red Button	(\$9.95)	www.adafruit.com/products/1185
PiT-Cobbler	(\$7.50)	www.adafruit.com/products/1989
Raspberry Pi Model A+	(\$24.95)	www.adafruit.com/products/2266
N.O. Pushbutton	(\$2.14/5pc)	http://amzn.com/B00D7C443C
Three-pin Connector	(\$5.28)	amzn.com/B00HR76YFE
4-40 Hex Standoffs and Hardware	(\$18.90)	http://amzn.com/B000GOIUUM
7" TFT-LED HDMI Display	(\$79.94)	http://amzn.com/B00L8VIHDW
Acrylic Sheet		(home store)
Enclosure and Decorative Materials		

PARTS LIST

could have been done with the camera module, but this was the best looking way to do it.

The time module is used to control timing on execution and to provide unique names for all of the photos based on the date and time they were snapped. (Time stamps are a great way to ensure unique file names — as long as your clock is right!) Raspberry Pi provides the GPIO library that lets us toggle and read pins for the buttons and LEDs. The Pi camera module controls the camera.

We start the live preview and do the photo captures with it. The sys and OS modules handle housekeeping tasks like cleanly exiting the program, shutting down the Pi, and working with the file system. To make things a little more fun, we use the Twython module to tweet the photos, and the random module to let us select a random set of Tweet text to go with the photos.

To start your photo booth, just run “sudo python photobooth.py” at the terminal. After a few seconds, your camera preview will appear. If it’s flipped, just exit the program (ESC key or Control C) and adjust the hflip or vflip settings in the code. Now, press the trigger button. The photo booth will turn on the power switch, do the countdown, take your photo, repeat three times, then turn the power switch off.

Make some faces and have fun, then exit the program. Your photos will be in the /home/pi/photobooth_photos directory and should be posted on Twitter if you have that enabled. If something went wrong, the error messages will point you to what part of the setup is incomplete.

Once you’re happy with how things are working, we need to set up the Pi to start up the photo booth program by default. We don’t want to connect a keyboard at every startup. Remember that easy setup was a requirement. We need the Pi to automatically log in and then start the script.

At a terminal, open the inittab file with your favorite editor (i.e., “sudo nano /etc/inittab”). Find the line that reads “1:2345:respawn:/sbin/getty 115200 tty1” and put a “#” at the beginning of the line to comment it out. Directly under it, type “1:2345:respawn:/bin/login -f pi tty1 </dev/tty1 >/dev/tty1 2>&1.” Press Control+o to save the changes and Control+x to close the editor.

Now, we’ll tell the Pi to run our program in the shell



■ **FIGURE 5. Careful removal of a viewfinder glass allowed me to mount an LED inside the camera and use the existing optics as an updated light pipe.**

Useful Links

SD Formatter
www.sdcard.org/downloads/formatter_4NOOBS

Download
http://downloads.raspberrypi.org/NOOBS_latestGitHub

Repository for PiBooth
<https://github.com/jrleeman/PiBoothPiBooth's>

Twitter — The Art of Command Line
<https://github.com/jlevy/the-art-of-command-line>



■ **FIGURE 6.** All electronics were mounted on an acrylic sheet as a backplane. Equipment with no mounting holes was hot glued or attached with double-stick tape. A cordless drill made quick work of the wiring harnesses.

profile. Open the profile in an editor (i.e., “sudo nano /etc/profile”) and at the bottom of the file, add “sudo python /home/pi/PiBooth/photobooth.py.” Reboot with “sudo reboot now” and you should be all set.

Enclosure

Now comes the real fun: designing your enclosure. You can get very creative here and build something to fit your exact vision. For me, I wanted a steampunk look that was durable. This meant wood, copper, antiques, and gears. I also couldn’t build the box from scratch since I didn’t have a table saw on hand.

Hobby stores and antique shops are a good place to go for inspiration and parts. I found a wooden cabinet that was unfinished for the main part of the system and a small metal box for the big red button. The boxes cost about \$20. At an antique shop, my fiancée found an antique Kodak “Brownie” camera for \$15. I repurposed the camera and brought it about 100 years forward in time with a digital upgrade.

I cut out a hole for the monitor using a jigsaw and drilled a hole for the IR receiver on the TV. The monitor was mounted with plumber’s tape, double-stick tape, and M4 screws (**Figure 4**). I stuck the remote inside the cabinet with some Velcro™ so it didn’t find its way into oblivion. All the cables were secured with hot glue. The front panel was now complete.

The Pi camera board was glued into the front part of the Brownie camera. I placed some electrical tape over the LED and used some black felt to conceal the green PCB (printed circuit board) from view. Then, carefully, I cut a slot in the paperboard part of the camera and threaded the flat-flex cable through. I bought a longer flat-flex cable, but this depends on your setup.

While the camera was apart, I removed some extra metal parts that were in the way, and also took out a glass plate from the viewfinder. That’s where my flashing LED was set with some glue. I used the jigsaw again to cut a



www.melabs.com

Development Tools and Learning Systems

PICmicro® microcontrollers





Reference Manual

PBP3

- PICBASIC PRO™ Compiler 3.0
- M.E. Labs Trainer
- U2 Programmer
- Prototyping Boards
- Experimenter Boards



Start learning programming today with M.E. Labs trainer board.

PICBASIC and PICBASIC PRO are trademarks of Microchip Technology Inc. in the USA and other countries. PIC is a registered trademarks of Microchip Technology Inc. in the USA and other countries.

slot in the top of my wooden box for the flat-flex. I then drilled two holes through the box and camera case to secure it down with some 6-32 hardware.

I mounted all of the electronics to an acrylic sheet using 4-40 hardware, standoffs, and hot glue. I then mounted that entire assembly in the box with the same hardware and standoffs (**Figure 5**). This backplane construction makes for easy removal/servicing and fewer holes in the box. Finally, I drilled a large hole for the mains plug and power switch plug to exit through, and concealed the space with more felt.

To dress things up a little more, I made some copper pipe handles out of 1/2" tubing and added a frame with instructions on how to use the photo booth. The frame and box were decorated with some gears and watch parts that are sold at hobby stores and on eBay.

The big red button fit very nicely in the diamond plate box I found (**Figure 6**). Step drills are the best tool for drilling large holes in panels, but be careful and take your time. I drilled a small hole in the corner of the box for the lead wire and fit it with a grommet. I used rotor wire since it was on hand, had three conductors, and was inexpensive. In the end, I didn't power the light in the button because it wasn't all that bright or well diffused.

Depending on your box, you may find it helpful to add some weight to keep it from skittering all around the table as people push it. I also used a rather overkill waterproof three-pin connector to let me separate the button and main unit for transport.

Closing Notes

You can fine-tune your photos by using the arrow keys to adjust the contrast and brightness settings for your venue. The up/down arrow keys control brightness and the left/right keys control contrast. These settings are reset with each reboot to the defaults that seem to work best in a normally lit room.

This photo booth provided our

guests with seemingly endless entertainment at the reception, and will likely be a hit at your gathering as well. If you wanted to make this a real booth with a curtain and a bigger screen, it is very scalable.

Other fun additions could include printing the photos for guests or adding an email option. The possibilities are endless. Your prop selection can grow and expand as you see what people have fun with, but we found the more over the top, the better. Happy photo taking! **NV**

HF-AUTO High-Power Analog Autotuner



“ My HF-AUTO is a superb piece of engineering magic
W9OWO Dec 2012

“ Beyond awesome I will never sell this fast, accurate, reliable
W2PP Sept 2015

“ Of the 30+ tuners I have owned ... the HF-AUTO is the best
K1ESE Feb 2012

The HF-AUTO is a microprocessor controlled fully automatic stand-alone tuner with a power rating of 5 Watts to 1800 Watts that will work with any transmitter built from the 1940s to the present. HF Bands: 160m to 6m. Three antenna outputs: SO239 coax. Dimensions: 12.5" W x 6.5" H x 16.5" D. Weight: 25 lbs. (11.4 kg).



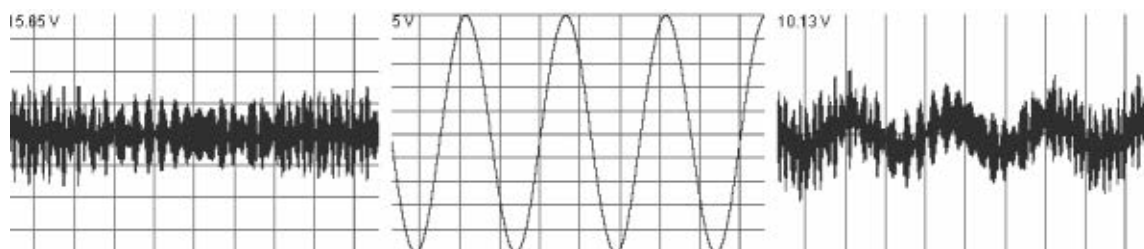
HF-AUTO-R Remote unit
Operate from 500' distance

Palstar product reviews from actual Hams as found on eHam.net

PALSTAR

Customer Support: 1-800-773-7931
Fax: 937-773-8003

www.palstar.com



Techniques to Improve Your Signal-to-Noise Ratio

What is noise? We all experience it and generally want to avoid it. Noise is usually an unwanted signal, and is the combined effort of every other signal trying to compete with one another. Think of when you are at a party or out at dinner in a restaurant and you have to talk louder just so that the people at your table can hear you. By talking louder, you have increased your signal-to-noise ratio by increasing the signal strength directly at the source. So, what if your output level or source signal is at a fixed volume or strength and you can't go any stronger? Then what?

One method is to repeat yourself so the person you are directing your conversation to can understand what you are saying. This amounts to transmitting the same data several times. This may improve the signal-to-noise ratio, but the rate of data then goes down as a consequence. What you have done and may not have even realized it is called ensemble averaging.

If you consider noise to be random in nature and it is combined with your true signal, then repeating yourself has a tendency to keep the random noise at a constant level while reinforcing the true signal embedded within the noise. For every time that you have to repeat yourself — or in terms of electronics,

“transmit” — your signal-to-noise ratio improves on the receiving end, and can be referred to as the square root of the number of samples “N” (**Figure 1**), or the number of times you said the same thing over:

$$\text{SNR} = \frac{N}{\sqrt{N}} = \sqrt{N} \quad \text{FIGURE 1.}$$

So, for example, if you had to repeat yourself twice, the signal-to-noise ratio would be 1.414-to-1. Likewise, if you had to repeat yourself three times, the signal-to-noise ratio would be 1.732-to-1. After the third time, I usually won't repeat myself with the exception of telling one of my daughters something I know they heard the first time. (That, however, is

another topic.) This method only works if you have the luxury of a signal that does repeat and the data rate is low. Using two or more sensors to receive the same signal can circumvent the need to read the signal over several times, but in practice may not be practical because of the physical redundancy in board real estate that the design requires.

There is a derivative of ensemble averaging that can be applied physically that avoids having to sacrifice the data rate called differential signaling. Instead of reading the signal twice, two sensors are used differentially to one another. This method has the ability to mathematically cancel out most of

your common mode signal (or noise) and provide a greater signal-to-noise ratio than just taking two samples and summing them together because while the signal increases, the noise floor is suppressed due to common mode rejection.

Honestly, I wish this method received more attention in some of the basic text books in schools. Many times, a book will cover a topic on how to read one sensor and simply move on without discussing the benefits one can gain from using two sensors together instead. This method is commonly used in data communication such as Ethernet, RS-485, etc., where the data being sent uses two signal lines that are 180 degrees out of phase. Differential signaling, however, is not limited to digital applications. It has been used for years in the audio industry, specifically for long lengths of microphone cables to balance out any electrical noise the cables — acting as an antenna — could pick up.

Differential signaling can be used for light sensors, sound applications, radio, accelerometers, gyros, and much more. For a thought exercise, think about what results you might get with two accelerometers mechanically locked together on a fixed plane and reading the signals from each of them differentially. As they move together, the signals would cancel out, but what if you pivoted one accelerometer on the Z axis with relation to the other one? What kind of signal would you see then? The signal would be similar to that of an angular rate gyro derived from using two accelerometers differentially that were mechanically locked to the same plane.

However you decide to use two identical sensors differentially, one sensor — if not used 180°

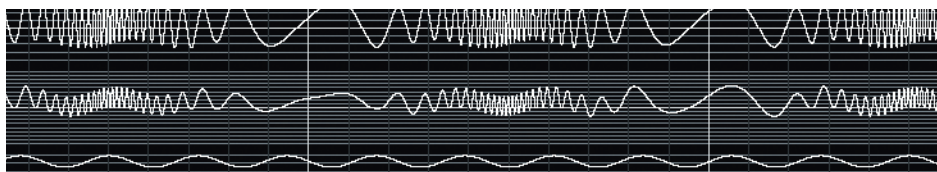


FIGURE 2.

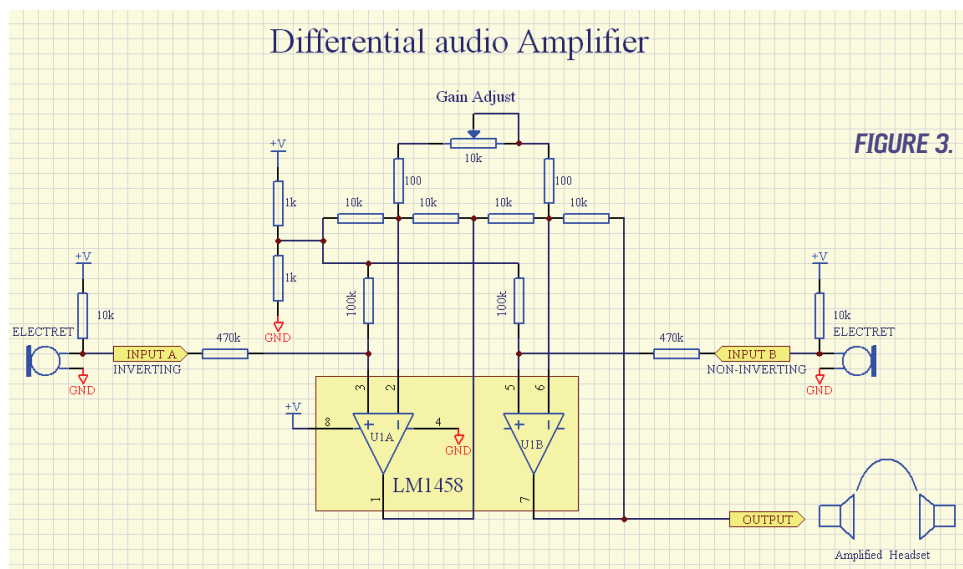


FIGURE 3.

out of phase — is used as a reference and the other sensor is used as the actual input. Referring again to the noisy restaurant scenario, slightly turning your head towards someone allows you to hear them better. We could argue the reasoning is that one ear is simply closer now to the source, and while some of that would certainly be true, your brain is a remarkable filter and there is more differential sensing and filtering going on than you might realize.

If the ambient noise is allowed to target both sensors (ears) simultaneously, then looking at the data from the sensors differentially will cause the ambient noise to

mathematically cancel out leaving only the true signal if one sensor is positioned slightly closer to the source than the other.

In **Figure 2**, the top and middle graphs represent the left and right channels with significant noise added. If you look closely, you can see slight variations between the two. The bottom graph in **Figure 2** represents the difference or differential between the left and right channels as a nice sine wave. Notice that all of the common mode “noise” present in the left and right was canceled out.

A practical circuit application of the above description would be for a

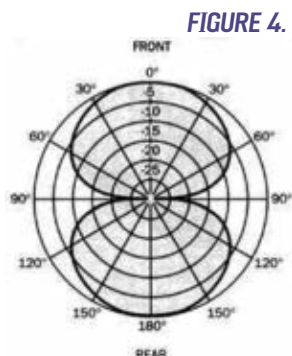


FIGURE 4.

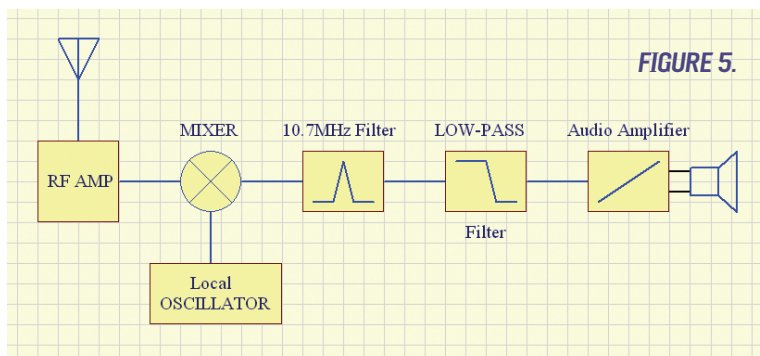


FIGURE 5.

microphone to block out any loud ambient noise, such as a plane or simply a windy day. The circuit in **Figure 3** will do just that. In this configuration, the LM1458 dual op-amp creates a differential input with a balanced load on each input. This way, the inputs electrically “see” the same load, which is important for

making this kind of differential measurement.

The microphones should be placed close to one another and will exhibit a pickup pattern that resembles a bidirectional microphone (see **Figure 4**) where one lobe is additive (non-inverting) and the other lobe is subtractive (inverting). In the

mid-region is a dead zone where the input from each microphone cancels out any common mode signals.

A method to improve the signal-to-noise ratio (adopted from early radio) is to use a regenerative circuit or Q-multiplier circuit. This method is essentially an active filter that reinforces itself with positive feedback. A small portion of the input signal is amplified and fed back into the input in a positive reinforcing way. This has the effect of attenuating any noise while at the same time re-enforcing a signal hidden within the noise. A transistor with a normal gain of 10 or so can have a perceived gain of 10's of thousands if applied in a regenerative circuit correctly.

Another method — again, adapted from early to more recent radio — is to use a local oscillator and mixer to level shift the input signal to another frequency band. This method is especially nice if you want to remove most (if not all) of the low frequency noise interference associated with 50/60 Hz, fluorescent lighting, etc., and create a final output signal with an exceptionally low noise floor.

Mixing is usually reserved if you have a particular target frequency that you are interested in. How it works is by literally mixing two signals together, which include the frequency that you are interested in plus a local oscillator frequency. Say,

ALL ELECTRONICS

www.allelectronics.com Order Toll Free 1-800-826-5432

22 GA. STRANDED HOOK-UP WIRE, 8 PC BUNDLE

Great for projects requiring color-coded wires. Bundle of 8 three foot lengths of 22 AWG stranded wire, assorted colors. Good quality wire, 300V 90°C, UL.

CAT# HW-24

100 for 40¢ each

2 for \$1.00



4-POSITION DUAL-ROW STRIP

Rated 20 Amps @ 300 Volts. Accepts wire from 22 to 14 AWG. 0.375" centers.

CAT# TS-204

\$1.35 each

25 for \$1.17 each
100 for 97¢ each



ULTRASONIC DISTANCE SENSOR

Can be used for object avoidance. The module has integral ultrasonic transmitters, receiver and control circuit and is for use in conjunction with Arduino boards. 45 x 27 x 17mm.

CAT# UST-4

\$4.50 each

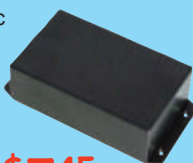


PROJECT BOX W/ MOUNTING FLANGE, 6.18" X 3.78" X 2.10"

High-impact ABS plastic utility boxes. Black textured finish. Interlocking lid with screws. Interior card guides for 1.6mm thick pc board. 6.18" x 3.78" x 2.10".

CAT# MBF-24

\$5.45 each



MINI-GRABBER TEST LEADS

Fully-insulated, spring-loaded 2" mini-grabbers. Flexible 24" leads with gold-flashed banana plugs. Set of red and black leads.

CAT# TL-9

\$5.95 each



18 AWG RED/BLACK ZIP CORD

Automotive grade "GPT" zip cord with red and black conductors. 80°C, flame-retardant PVC insulation. Available in rolls of 25, 50, 100, 250, 500 and 1000 feet. We will cut to order, 10 ft. minimum.

CAT# WRB-18

25' for \$8.00

50' for \$15.00 • 100' for \$23.00



GEARHEAD MOTOR W/ WORM GEAR LINEAR ACTUATOR

A great motor for linear movement applications. 5.9:1 ratio gear motor with 9" worm gear and drive sleeve. The rotary motion of the screw is translated to linear motion by the drive sleeve. Reversing the motor, reverses the direction of movement. Rated 24Vdc. 2.0" diameter (gear box) x 3.7" long.

CAT# DCM-479

\$27.95 each



15' MICRO-USB CABLE

15' cable for charging & data on cell phones, PDAs and digital cameras. Standard USB-A to Micro-USB-B connector.

CAT# CB-434

\$4.00 each



References

Signal-to-noise ratio

http://en.wikipedia.org/wiki/Signal-to-noise_ratio

Shot noise

http://en.wikipedia.org/wiki/Shot_noise

Ensemble averaging

http://en.wikipedia.org/wiki/Ensemble_averaging

Regenerative circuit

http://en.wikipedia.org/wiki/Regenerative_circuit

Microphone basics

www.ccisolutions.com/StoreFront/category/microphone-basics-part-2

for instance, that you want to listen to 100.5 MHz on the FM dial (refer to **Figure 5**). Without knowing it, you are actually tuning the local oscillator to 89.8 MHz ... the reason will be explained later.

When you combine two frequencies, you get those two frequencies, but you also get their sum and difference derivative frequencies. So, for 100.5 MHz and 89.8 MHz, you not only get those two frequencies, but you also get 10.7 MHz and 190.3 MHz. The radio industry has standardized a specific filter frequency for both FM and AM radio which happens to be 10.7 MHz for FM and 455 kHz for AM. Because the example radio station is FM, the 10.7 MHz filter is of importance here.

What this filter does is eliminate everything but the 10.7 MHz frequency, allowing only that frequency to pass. Since that frequency is only produced in the receiver circuit if your target frequency is present and agrees with the local oscillator frequency, it can be said that if 10.7 MHz is present or detected at all, then you are “tuned” to your target frequency or that the target frequency is present somewhere within the received signal.

Next — and this seems counter-intuitive — is that we remove the 10.7 MHz signal we just tuned to by using a low-pass filter. What this does is not only eliminate the 10.7 MHz signal we just filtered, but by passing it through a low-pass filter we are left with the low frequency modulated data originally transmitted from the radio station. This can be audio data, digital data, etc. For standard FM radio, there are a few more filtering processes that need to take place, such as decoding the left and right stereo, etc. Hopefully, the idea here is more on how to detect and isolate your signal from the noise. A similar technique can be applied to ultrasonic, IR, etc., and does not necessarily need to be confined to radio. I have used this technique on

RFID tags and some ultrasonic applications, as well as rolling my

own IR receiver for a non-standard modulation frequency. **NV**



probotix CNC Routers & Control Systems

When You Are Serious About CNC

www.probotix.com

A QR code is located in the top right corner of the advertisement.

Make up to \$100 an Hour or More!



**Be an FCC
LICENSED
ELECTRONIC TECHNICIAN!**

Get your “FCC Commercial License” with our proven Home-Study Course!

- No costly school. No classes to attend.
- Learn at home. Low cost!
- No previous experience needed!
- **GUARANTEED PASS!** You get your FCC License or money refunded!



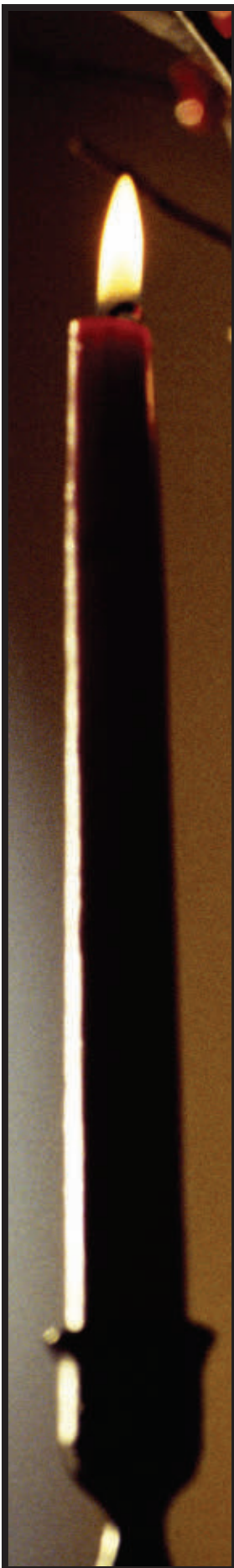
Your “ticket” to thousands of high paying jobs in Radio-TV, Communications, Avionics, Radar, Maritime and more.

Call for FREE info kit: 800-877-8433 ext. 109

or, visit: **www.LicenseTraining.com**

COMMAND PRODUCTIONS • FCC License Training
Industrial Center, 480 Gate Five Rd., PO Box 3000, Sausalito, CA 94966-3000





Make Your LED Dance Just Like a Candle

Looking to add some dynamic lighting to your room? Then, modulate an LED to respond to the world around it with this simple technique.

Light emitting diodes (LEDs) are the perfect electronic transducer. They convert electric current into light with typically a 10%-30% efficiency, and can respond in sub-microsecond time scales. Their brightness and efficiency is why they are becoming the standard for lighting applications. They've already taken over the flashlight and portable application space, and as their price goes down, will likely take over the residential light bulb application space as well.

However, if all we do with an LED is use it for steady, dull, boring illumination, we are leaving so much of their performance on the table. Instead, why not take advantage of their ability to quickly respond to a changing current, and modulate their drive current to turn your room into a responsive environment?

In this article, I'm going to show you how to make any LED dance to the small randomly unstable air currents that make candles flicker. The technique can be applied to turn ANY environmentally-sensed input into an LED's or light bulb's brightness.

The usual 10 mA current levels recommended for LEDs just aren't bright enough for me, however. I want brighter. How bright, you ask? As my buddy, Frank Schonig says, "A lot is good, more is better, and too much is just right." I'll show you how to modulate a current source right up to the limits of "too much."

The Basic Architecture for this Project

There are lots of ways of turning a sensor's signal into an LED's light output. In this project, I chose to use an Arduino as the go-between because I wanted a general approach, and know I can add additional scaling and adjustments to the sensor signal using a little bit of code running in the Arduino. As good as I am in analog circuit design, I know I can get to the final performance I want faster and easier with

the advantage of some digital signal processing.

The architecture of the system I wanted to implement — which I think is a good template for many applications — is shown in **Figure 1**. The start is the sensor which is picking up a particular quality in the environment and turning it into an electrical feature: a current, voltage, or resistance.

The interface between the sensor and the Arduino is the Analog Front End (AFE) that turns the sensor response into a high level voltage signal the Arduino's

Analog-to-Digital Converter (ADC) can read. This is usually some combination of resistors, voltage bias, op-amps, and instrumentation amplifiers.

Once the digital signal is in the Arduino through its ADC, we can apply the power of math with digital signal processing to turn these bits into the appropriate Pulse Width Modulated (PWM) voltage signal that will drive adjustable current through the LED. There is a limit to the output current of an Arduino pin. If we need more current, we can use a current booster. This can be as simple as a transistor follower circuit.

To illustrate how to implement each section, I'll walk you through three examples which go from simple to more complex:

Example #1: Internally generated sine wave signal and Arduino limited LED brightness.

Example #2: Pushing the ultimate current limits to an LED.

Example #3: Modulating the LED brightness with an air current sensor.

Example #1: Modulating an LED's Brightness: The Quick and Simple

In this first example, we're going to modulate an LED's brightness with a simple math function generated inside an Arduino. I chose to use a sine wave as the example. If you can describe the function as an intensity vs. time, you can use it to modulate the LED or any other light source.

Every sine wave has just three terms that characterize it: the amplitude, frequency, and phase. The phase just says where in the cycle we start the sine wave. To make it easy, we'll use a phase of 0. Then, it's just the amplitude (A) and frequency (f) we need to define.

Sine waves are bipolar, rising equally above and below zero in magnitude. **Figure 2** shows the calculated sine wave signal with an amplitude of one and a frequency of 1 Hz. The pattern repeats every 1 sec, so its frequency is one cycle per second — or 1 Hz — in this example. We have to modify this sine wave slightly to use it to modulate an LED. While the sine wave magnitude drops below zero, an LED's light intensity cannot. Its lowest value is 0. We're going to shift the sine wave up so it extends between 0 to the peak value, A. The sine wave function for an intensity $I(t)$ is:

$$I(t) = \frac{A}{2} (\sin(2\pi ft) + 1)$$

The factor of 1/2 at the beginning scales the sine wave so that its maximum value is $A/2 \times 2 = A$.

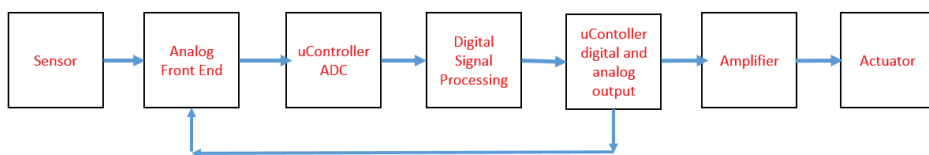
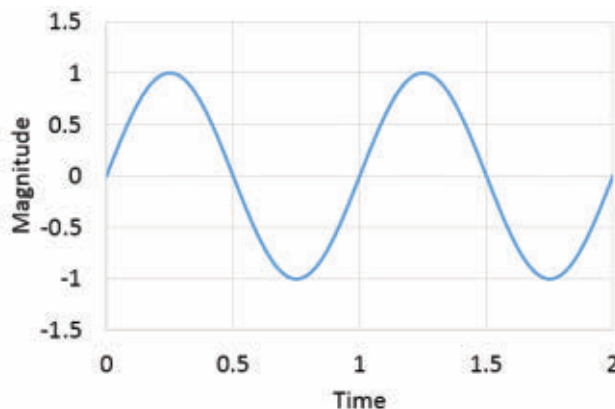


Figure 1. Flow diagram of the system implemented in this project, which can be used for many similar applications.

Figure 2. Example of the magnitude of a sine wave with an amplitude of one and frequency of 1 Hz, showing the signal going above and below zero.



This is literally the equation we will use in the Arduino sketch to modulate the LED. This will be the analog voltage output to the Arduino Digital-to-Analog Converter (DAC) pin which will drive the LED.

Analog Output Signals and the Arduino

The Arduino and many other microcontrollers don't really have an analog output signal. Instead, they have a PWM signal. In the Arduino, it is a 500 Hz digital signal with a duty cycle that is modulated. The average of the modulated signal is the analog voltage.

When the `analogWrite` value is set to a high level, the duty cycle is nearly 100%. When set to a small value, the duty cycle is close to zero. **Figure 3** shows the measured PWM output voltages on pin 3 for an output level that is a 4.5V/90% duty cycle, and a 0.5V/10% duty cycle signal.

Not all the Arduino pins can be used for PWM output. On my RedBoard from SparkFun, the PWM pins are identified with a "~" (tilde) symbol written right on the

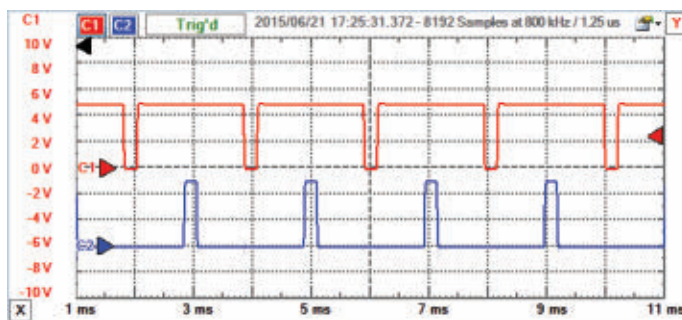


Figure 3. Measured output voltage of two PWM signals. Top: set for 4.5V output. Bottom: for 0.5V signal. The frequencies are the same 500 Hz, or a period of 2 msec. The duty cycle of each signal is adjusted to control the average value.

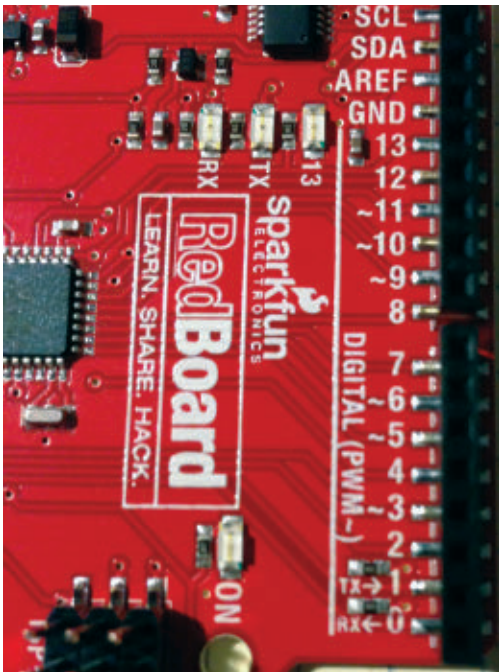


Figure 4. Closeup of the RedBoard from SparkFun showing the PWM capable digital I/O pins marked with the “~” symbol on the right side of the board.

board. A close-up of my board is shown in **Figure 4**. On this RedBoard, digital pins 3, 5, 6, 9, 10, and 11 are capable of PWM.

To use pin 3 as an analog output, we need to set up the pin in the *void setup()* function, using *pinMode(3,OUTPUT);*, for example. Then, in the section of code to output the analog voltage, we use the line *analogWrite(3, 125);* where the number 125 is an example of an integer from 0 to 255 — an eight-bit number.

Normally, if we wanted a true analog signal from a PWM pin, we would have to add some sort of RC circuit to integrate the pulses and turn it into an averaged nearly DC signal. When driving an LED, however, the PWM signal is ideal. Our eyes do the averaging. We can’t resolve the 500 Hz pulses unless we move our eyes

```
72 void drive_sine_wave() // modulate LED with a sine wave
73 {
74   float time_now_ms = (millis() - timer0); // this is the time in sec from timer start
75   float time_now_sec = time_now_ms / 1000.;
76   int Int_LED1_8bits = Amp_LED1_8bits / 2 * (sin(2 * pi * freq_Hz * time_now_sec) + 1);
77   analogWrite(6, Int_LED1_8bits); // driving LED with pin 6
78   Serial.print(time_now_sec);
79   Serial.print(" ");
80   Serial.println(Int_LED1_8bits);
81 }
```

Figure 5. Code snippet which generates the sine wave signal and outputs to the PWM pin 6. This function is called in the *void loop()* routine to drive the LED with a sine wave. The serial print commands are there for debug.

around very fast. Then, we see the pulses of light in our after-image.

This is one way of telling if a car’s lights are LEDs or a light bulb. Next time you are behind a car at night, move your eyes around quickly and look for a train of pulses or a solid line in the wake of the light source. The typical recommendation for hooking up an LED to an Arduino pin is to use a 330 ohm resistor in series. For this initial example, we’ll follow this simple recommendation.

All the code that runs in the *void loop()* to drive the LED with a sine wave modulation is shown in **Figure 5**; the close-up of the whole circuit is shown in **Figure 6**.

When the code runs, a 1 Hz sine wave with a peak-to-peak value from 0 to 255 with eight-bit steps is written to the PWM pin. The actual eight-bit number written to the pin was also printed to the serial port; I used MakerPlot to plot this number just to verify the signal. **Figure 7** shows the plot of this value over time.

Example 2: Driving the LED with “a lot of Current”

If you read the spec sheets for most LEDs, they say use 10 mA of drive current and do not exceed 20 mA. Really? How much current can we really put through an LED, and how do we do it?

An LED behaves like a really low resistance circuit element but with a constant voltage drop across it. Depending on the LED’s color, this voltage drop varies from about 1.6V to 3.5V. For a red LED, it’s about 2V with any current more than 10 mA through it.

This means that the current through an LED is not so much about the LED, but about the rest of the circuit it is in. When connected to the output pin of an Arduino, the equivalent circuit looks like the one in **Figure 8**.

From this, we can calculate the current in the whole circuit and through the LED:

$$V_{\text{open}} = I R_{\text{pin}} + I R_{\text{load}} + V_{\text{LED}} \quad \text{and}$$

$$I = \frac{V_{\text{open}} - V_{\text{LED}}}{R_{\text{pin}} + R_{\text{load}}}$$

Let’s put in some numbers to see the special cases. For a red LED, we’ll use 2V as the LED voltage drop. I’ve measured the output impedance

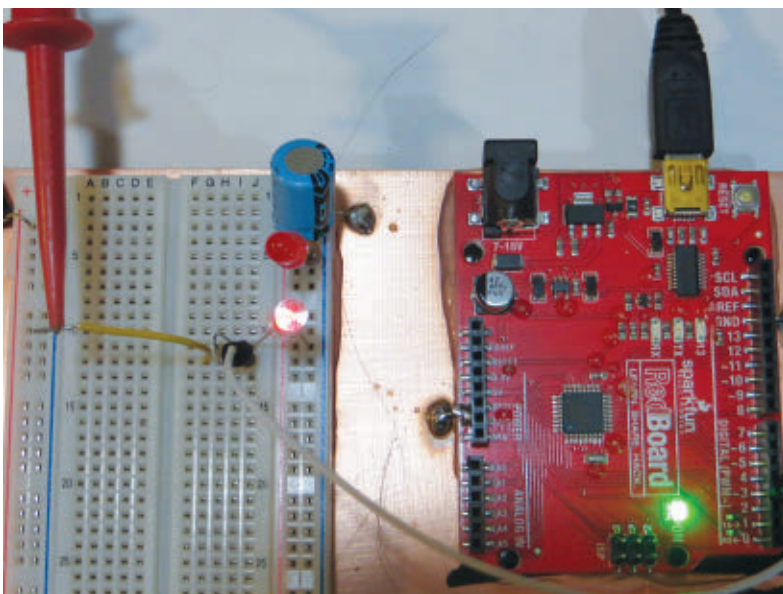


Figure 6. Closeup of my Arduino circuit with the LED connected to pin 3 with a 330 ohm resistor. The clip lead is connecting the voltage from pin 3 to the scope.

of an Arduino pin as about 25 ohms. The open circuit voltage of a pin with nothing attached is about 5V. When we use a 330 ohm load resistor, the current through the LED is:

$$I = \frac{V_{\text{open}} - V_{\text{LED}}}{R_{\text{pin}} + R_{\text{load}}} = \frac{5V - 2V}{25 + 330} = 8.5 \text{ mA}$$

This is close to the 10 mA typical value suggested and why 330 ohms is often recommended. The ultimate limit to the current we can get out of an Arduino pin is if we didn't use any load resistor at all. We'll still have the 25 ohm output impedance of the pin electronics to limit the current. Let's see how much current we'd get if we did not use the 330 ohm resistor:

$$I = \frac{V_{\text{open}} - V_{\text{LED}}}{R_{\text{pin}} + R_{\text{load}}} = \frac{5V - 2V}{25} = 120 \text{ mA}$$

This sounds like a lot – and it is – compared to the spec sheets. Just to verify, I was really getting 120 mA from the Arduino pin. I added a one ohm series resistor to the LED and measured the voltage across it. I measured about 100 mV across a one ohm resistor, so there was 100 mA of current through it. This is slightly lower than our estimate because the voltage drop across the LED is a little higher than 2V, and the output impedance of this Arduino pin is probably slightly higher than 25 ohms.

The spec sheet for the Arduino Uno says the maximum current draw per pin is 40 mA, with 200 mA as the maximum total current draw for all the pins. This is just the safe recommended limit. Each pin can source more current than the specified value, but we run the risk of damaging the Arduino.

With 100 mA through our LED, it's pretty bright. A lot is good, but more could be better. To drive the LED with more current, we need a current booster circuit.

A Current Boost Circuit: “More is Better”

There are a number of ways of driving more current

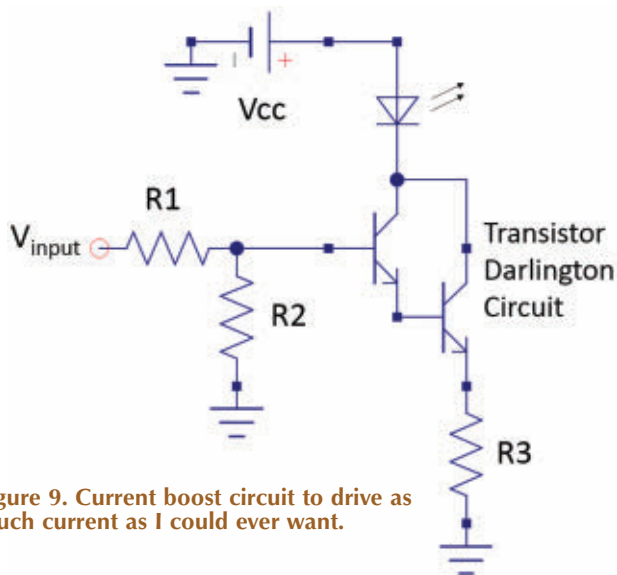


Figure 9. Current boost circuit to drive as much current as I could ever want.

Figure 7. Plot of the eight-bit value written to the PWM pin and to the serial port, and displayed using MakerPlot.

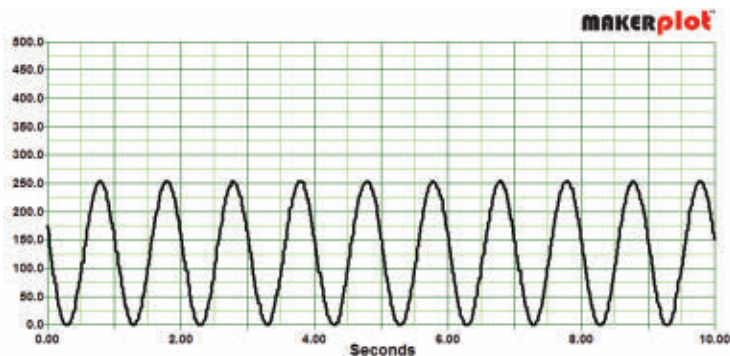
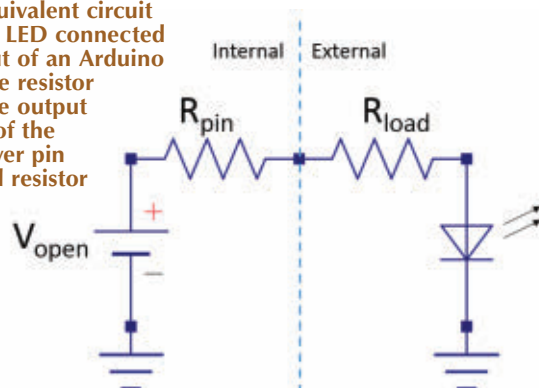


Figure 8. Equivalent circuit model of an LED connected to the output of an Arduino pin. Note the resistor modeling the output impedance of the Arduino driver pin and the load resistor we add to the circuit.



from a fixed current source. The simplest one I chose is a transistor follower circuit with the LED between the Vcc power source and the collector of the transistor. This circuit is shown in **Figure 9**.

A transistor is a current amplifier. The current flowing from the collector to the emitter is a factor times the current into the base. This factor is usually designated as h or h_{FE} . For the TIP31B transistor I am using, the value of h is a minimum of 25, but could be as high as 100. This gain was a little smaller than I wanted but I had plenty of the TIP31B transistors lying around, so I decided to use them in a Darlington configuration which is shown in the circuit.

The current gain of these transistors in series is the product of the gain of each one. In this circuit, the current into the base is amplified by at least $25 \times 25 = 625$. This transistor is rated for 3A DC current. I figured the LED would burn out long before the transistor would. In this circuit – called a follower circuit – the current through the transistor from the collector to the emitter and through the LED will automatically adjust to keep the emitter voltage plus the base-emitter diode drop voltage to be the base voltage.

The first two resistors, R1 and R2, act as a voltage divider. The output voltage from the Arduino pin is always an amplitude of 5V. It's the duty cycle that is modulated. I wanted to make the maximum voltage into the transistor's base adjustable to limit the maximum current through the LED. The current through the transistors will automatically change so that the voltage across the base is the two diode drops of the two base-to-emitter junctions, in series with the IR drop across resistor R3:

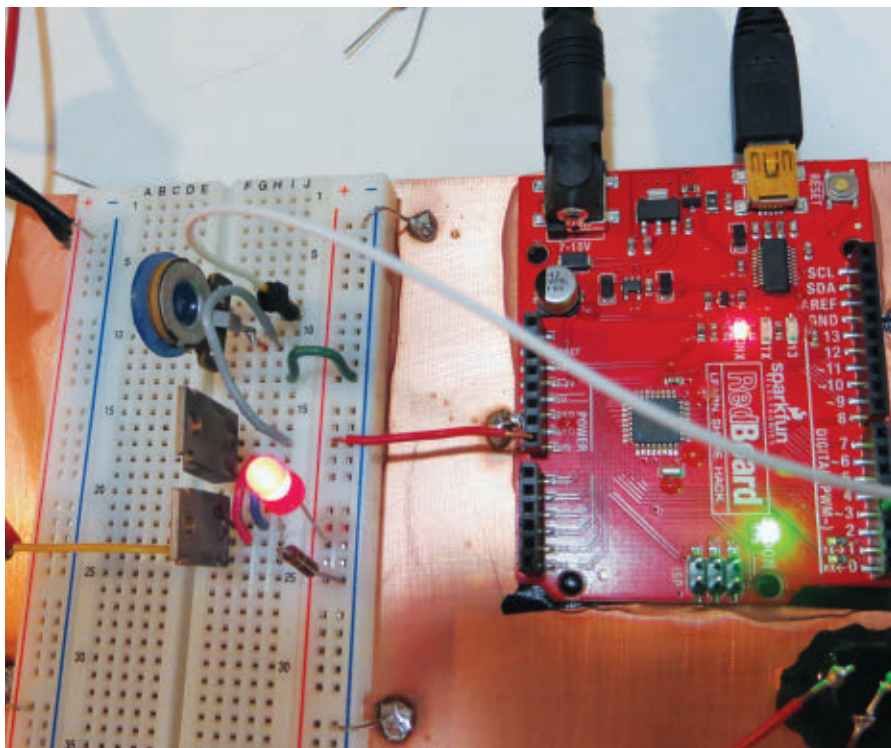


Figure 10. The setup using my Darlington current booster driving an LED from an Arduino pin.

$$V_{\text{input}} \frac{R_2}{R_1 + R_2} = (0.6V + 0.6V) + I \times R_3$$

We can use this to determine what load resistor to use based on the maximum current we want to reach:

$$R_3 = \frac{1}{I} \left(V_{\text{input}} \frac{R_2}{R_1 + R_2} - 1.2V \right)$$

If we want the maximum current to be 0.5A, when the voltage divider is set to the maximum voltage so R_1 is nearly 0 and the output voltage is 5V from the pin, the load resistor, R_3 , should be about:

$$R_3 = \frac{1}{0.5A} (5V - 1.2V) = 7.6 \text{ ohms}$$

I used a 10 ohm resistor because I had one laying around. Whenever we use low resistance and 5V, we always want to pay attention to the power consumption to make sure we won't generate smoke, unless that's our intent. In the worst case — with 0.5A through the 10 ohm resistor — the power consumption will be:

$$P = I^2 R = 0.5^2 \times 10\Omega = 2.5 \text{ watts}$$

I only had a half watt resistor. This meant there was the potential for the resistor to get very hot and maybe smoke. I had to watch out for this. Of course, I expected my LED to die long before I got to 0.5A. When an LED dies, it usually dies as an open, so it would limit the current. The last estimate we need to do is for the resistor divider to scale the Arduino's output signal into a maximum current. If we really want 0.5A through the transistor and we have a current gain of only 625, then we need to supply at least $0.5A/625 = 0.8 \text{ mA}$ of current into

the base of the transistor. With a 5V source, the resistance of the voltage divider would have to be no more than $R < 5V/0.8 \text{ mA} = 6K \text{ ohms}$. No problem. To provide easy adjustment, I used a 1K pot. A 10K pot would probably have worked just as well.

Finally, I had a 9V switched mode power supply (SMPS) from SparkFun which was able to supply 1A of current at 9V. I used this to power the Arduino and picked off this voltage to power the LED. **Figure 10** shows the setup capable of driving 0.5A of modulated current through my LED.

Just How Much is "Too Much?"

By adjusting the 1K pot, I could adjust the maximum voltage across the base of the Darlington, which adjusted the maximum current through the LED as the Arduino produced the PWM sine wave signal which modulated the LED. As I cranked up the maximum input voltage to the transistor's base, I monitored the peak voltage across the

10 ohm resistor with my scope. This voltage is a direct measure of the current through the LED.

At 160 mA of current (1.6V across the 10 ohm resistor), the red LED I was using started to not turn off completely at the bottom of the sine wave cycle. This was probably some thermal effect in the LED. At 200 mA, the LED was mostly on and did not turn off at all at the bottom of the cycle. Plus, it was a bit warm to the touch.

Using an ultra-bright red LED and 160 mA of current, the LED was too bright to look at directly. After the after-image cleared, I cranked the current up. Beyond 200 mA, the intensity seemed to drop, and the LED was warm to the touch.

I also tried a bright white LED. One of the cool features of this current boost circuit is that it has plenty of voltage head room to drive any color LED — no matter what its forward voltage drop. At 150 mA of peak current, the white LED was brilliant. At 200 mA, it began to fade and the fade was irreversible. I damaged the LED.

This exercise defines the limits to driving most LEDs in a near steady state. The maximum brightness is with about 150 mA of current, and 200 mA is too much and will damage the LED. For most LEDs, the current limit is really set by a thermal limit. Running an LED at 200 mA with a 2V forward voltage drop is a power consumption of $2V \times 0.2A = 0.4 \text{ watts}$. This is dissipated in a very small volume and thermally insulated from the outside in its plastic housing. It's easy to imagine the diode getting too hot and being thermally destroyed. If all I wanted was pulses — like for a strobe — I could probably get away with more current if it lasted for a short time. However, I was going to run this LED as mostly steady on for long periods.

Even running at 150 mA, I am severely decreasing the lifetime of the LED, but like Achilles, I'd rather my LED have a short but bright and glorious life, than a long life at a dull intensity. To make my life easier, I decided to run at 150 mA and use a socket for the LED, so I could easily replace it should the need arise.

Example 3: Dancing to Thermal Fluctuations

The original idea I had for this project was to modulate an LED with small air currents and have it behave like an electronic candle. The perfect air current sensor is a warm thermistor which is self-heating by its own

current. I've used this in many similar sensor applications. The resistance of most conductors increases with temperature. This is fundamentally related to the scattering of the electrons by the vibrations of the crystal lattice of the metal. The higher the temperature, the more the lattice vibrates, the more the electrons are scattered, and the higher the resistance. For copper, the resistance increases about 0.4% per °C. This is called the temperature coefficient of resistance (TCR). It is positive in most metals.

In conductive ceramic materials, the TCR is negative. Increasing the temperature decreases the resistance. They are called negative temperature coefficient (NTC) materials. In these materials, the number of conductors released into the material increases with temperature. At higher temperatures, the material becomes more conductive. The resistance and temperature is highly nonlinear, so we don't often use a TCR value, but can easily see 100% changes over ambient temperature ranges. These make excellent temperature sensors.

My idea was to use a very small size "bead" type

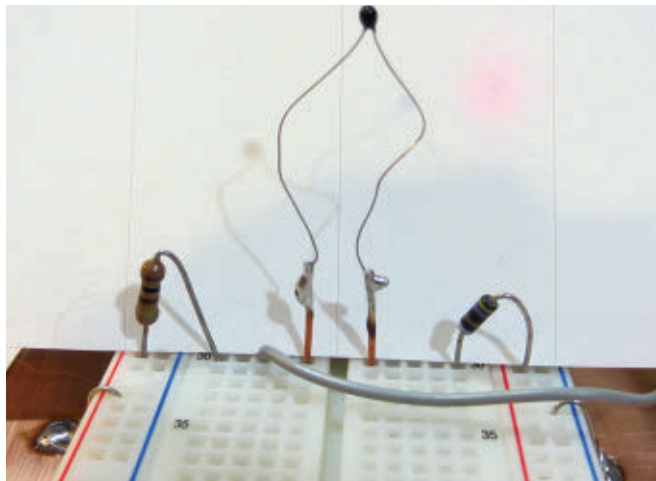


Figure 12. Closeup of the bead thermistor with its very fine wires soldered to copper leads mounted in the proto-board with the two bias resistors.

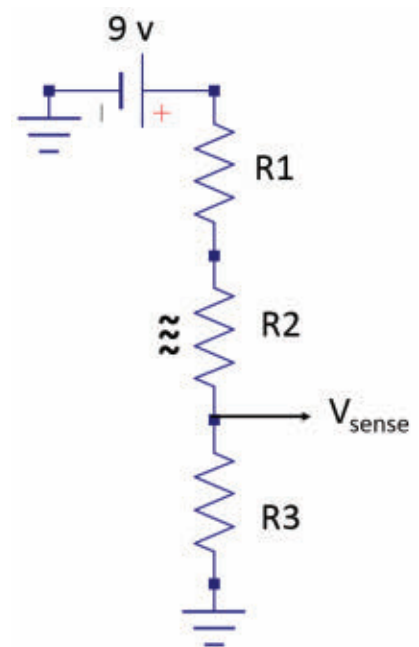
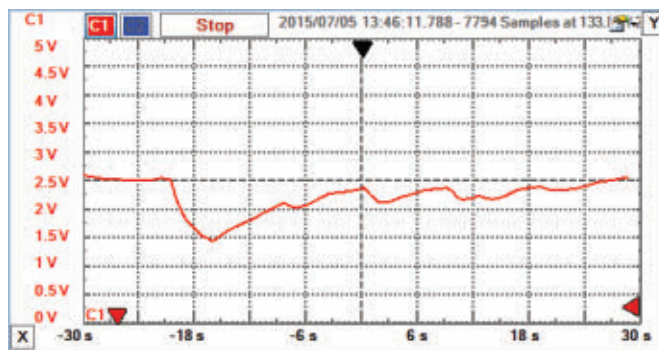


Figure 11. Schematic of the thermistor biasing circuit. R1 is 50 ohms, R2 is the thermistor, and R3 is the 100 ohm sense resistor.

Figure 13. Scope recording over a one minute period of the voltage across the sense resistor when the thermistor was touched, cooling it and lowering its temperature. This shows the range of the voltage signal spanning 1V.

thermistor with a fast response time which I could heat above ambient. As small air currents cooled it, its temperature would decrease and its resistance would change. I would modulate the LED's current based on the resistance of the thermistor. An example of a suitable thermistor is the MF52 series from Cantherm (a Canadian company) which costs less than 50 cents each.

The original idea was to use a simple voltage divider circuit with the thermistor as one leg of the divider. I already had a 9V supply in the system. This would be plenty to power the thermistor. The power dissipation in a resistor is V^2/R . For at least 0.1 watts of power in the thermistor, to get it warm I would need no higher than a 1K ohm thermistor.

One danger to watch out for is just attaching the thermistor across a constant voltage supply. The current through it would heat it up. With its negative temperature coefficient, its resistance would decrease and more current would flow through it. It would then dissipate more power and get hotter. Its resistance would drop and the current would increase and ... you get the idea. This is called thermal runaway.

To avoid this problem, I added a series resistor to create a voltage divider. This enables a simple way of measuring the resistance of the thermistor and limiting the current to prevent thermal runaway. **Figure 11** is the final schematic I settled on.

The thermistor is a small bead on the end of very fine

Thermistors:
www.cantherm.com

MakerPlot:
www.makerplot.com

SparkFun RedBoard:
www.sparkfun.com/products/12757

Digilent Analog
Discovery Scope:
www.digilentinc.com/Products/Detail.cfm?NavPath=2,842,1018
&Prod=ANALOG-DISCOVERY

Resources



Figure 14. Measured voltage across the sense resistor in still ambient air, showing variations on the order of 50 mV. This is the sort of fluctuations that will emulate a candle flicker, and will be used to drive the LEDs.

wires. This helps to thermally isolate the sensing element. **Figure 12** is a closeup of the thermistor soldered to copper wires, and the other resistors in the protoboard I used to assemble the entire circuit. When I first tried the thermistor — which had a nominal resistance of 1K ohm at low current — I thought a 1K resistor would be an ideal sense resistor. However, when I connected this circuit, the thermistor went into a thermal runaway and its resistance dropped to nearly 200 ohms. I finally settled on a 100 ohm sense resistor. Using the 9V supply to bias the thermistor would have resulted in a nominal voltage of about 4.5V on the sense resistor. I wanted this closer to 2.5V — the sweet spot for the Arduino's ADC.

The final circuit used a 100 ohm sense resistor and a 50 ohm resistor to shift the nominal sense voltage closer to 2.5V. This means the total resistance in the path is about 250 ohms. The current draw through the thermistor was $9V/250\text{ ohms} = 36\text{ mA}$. With this current, the power dissipation in the thermistor is $I^2 \times R = 0.035^2 \times 100\text{ ohms} = 125\text{ mW}$. This power is enough to heat the tiny bead thermistor to be hot to the touch. This is exactly what I wanted.

The slightest air motion is enough to cool the tiny bead thermistor and lower its temperature. This increases its resistance, and the total current in the circuit decreases. The voltage across the sense resistor will also decrease. **Figure 13** is a scope trace of the voltage on the sense resistor in still air and then when I touched the thermistor to cool it down. In this example, we can see the maximum

voltage range of the sense resistor spanning about 1V. This is well inside the range of the ADC.

The real fluctuations from tiny air movements are very small. **Figure 14** shows the measured voltage across the sense resistor from still air motions cooling the thermistor, which are less than 100 mV in extent. The challenge left is to turn the small fluctuations into a large enough signal to drive the LED so it mimics the flickering of a candle.

A Little Digital Signal Processing

I needed to turn these small voltage fluctuations into higher level signals to drive the LED current. I experimented with an AD623 instrumentation amplifier front end. This worked really well. I could amplify these low level signals into any range I wanted. However, the dynamic range of the signal was a 1V variation. If I used a gain of more than five, I ran the risk of saturating the output signal.

However, if I used another Arduino DAC pin to create an offset voltage which I could adjust, I could always change the offset voltage so the difference signal was in range. I built all this up and it worked just fine, and then realized there was a simpler way.

These 50 mV sensor fluctuations I wanted to be sensitive to are just in the range that can be resolved with a 10-bit ADC. The resolution of a 0V to 5V 10-bit ADC is 5 mV.

Even though the instrumentation amplifier AFE gave me unlimited headroom, I decided to go for a straight up digital signal processing solution. I wanted to use only the fluctuations of the sense resistor voltage to drive the LED. Yet again, if I just took a scaled version of the fluctuations, I would quickly reach the voltage limit of the DAC and saturate the LED.

So, I used a trick of re-establishing the baseline. I initially measure the voltage on the sense resistor. This is stored as a reference value and subtracted from every consecutive value. This difference is scaled by a factor and then centered about an offset value. This signal is then used to drive the LED intensity.

However, I also monitor the voltage level to the LED. If it is close to saturation and exceeds a predefined limit — either too high or too low — I take another reading on the baseline and re-center the LED drive voltage. This continually shifts the LED drive voltage so it is always in range, and allows any scale factor or sensitivity. **Figure 15** is the code snippet which executes this simple algorithm.

Nominally, I wanted the roughly 100 mV of sensor excursion to be mapped into a 0V to 5V output signal. This is a scale factor of about 50. I experimented with different scale factors and found a value of from 30 to 50 seemed to give the electronic candle its most realistic performance. **Figure 16** is an example of the typical signal which drives the LED when the sensor is in still air. This is the eight-bit signal that is written to the PWM pin. I also wrote this value to the serial port and used MakerPlot to plot this data.

With a scale factor of 30, the 250-bit range

```
43 void candle() // modulate LED with input from A1
44 {
45   // RawSensor_10bit is measured in the void setup ()
46   Adjusted_10bit = ((analogRead(A1) - RawSensor_10bit) * Scale_AD + Offset_AD);
47   if (abs(Adjusted_10bit - 512) > ThresholdReset_10bit) {
48     RawSensor_10bit = analogRead(A1);
49   }
50   DAC_8bit = Adjusted_10bit / 4;
51   analogWrite(6, DAC_8bit); // driving LED with pin 3;
52   //Serial.println(Adjusted_10bit);
53   Serial.println(DAC_8bit);
54 }
```

Figure 15. Code snippet to re-center the LED signal each time it exceeds a pre-set threshold.

Figure 16. Plot of the eight-bit signal to the PWM pin that drives the LED when the thermistor was sitting in ambient still air. The fluctuations in slight air movement from the hot thermistor drive the LED signal. Note the re-centering happening at 17 seconds, when the voltage exceeded the upper threshold, and at 49 seconds when the signal exceeded the lower threshold.

corresponded to about $5V/30 = 167$ mV. When I touched the thermistor, the voltage could change by as much as 1V, which is actually 12 re-centering operations. Figure 17 shows the signal to the LED when I touched the thermistor. There were about 10 very rapid re-centerings as it cooled to my touch, and then about eight re-centerings in the upward direction as it warmed back up. These very nicely emulated the rapid flickering of a real candle.

This simple algorithm of high gain to capture the fluctuations and an auto re-centering of the signal gave my electronic candle the flickering characteristics of a real flame. I could gently blow in the direction of the thermistor and watch the LED flicker as the sensor cooled and then warmed back up. In steady air, the self-heating of the sensor generated its own convection air currents and always kept the light flickering with a slow and seemingly random pattern just like a regular candle.

An Even Better Solution

I set out to build an electronic version of a candle that would respond to the gentle fluctuations of air currents, and modulate an LED to emulate the hypnotic allure of a fluctuating flame. I found that a self-heated thermistor is the perfect sensor, and the Arduino has just barely enough sensitivity in its ADC to take the small voltage fluctuations from the sensor and scale them into the range which makes an LED behave like a candle. A simple algorithm always keeps the LED signal in range and adds the feature of rapid flicker.

Along the way, I found a simple circuit which boosts the low current from an Arduino PWM pin to enable me to drive as much current as I could ever want. I originally thought this would be great for driving an LED. However, the circuit and system works for any current driven light source that doesn't need more than about 4V of bias.

Even though I am enamored with LEDs, for this application I found a simple incandescent light bulb gave an even more dramatic effect. With this 9V SMPS, I was limited to using a light bulb that took less than 4V to turn on. Just replacing the 9V supply for a 15V supply allowed me to use a 6V night light bulb. This is one of the advantages of this specific current boost circuit, with the light source between the voltage source and the transistor's collector.

In the circuit, the light bulb fluctuates as though it were a real candle. It flickers and responds to the subtle air currents around it. In a small dark room, the flickering of the light bulb really does look exactly like an electronic candle. **NV**

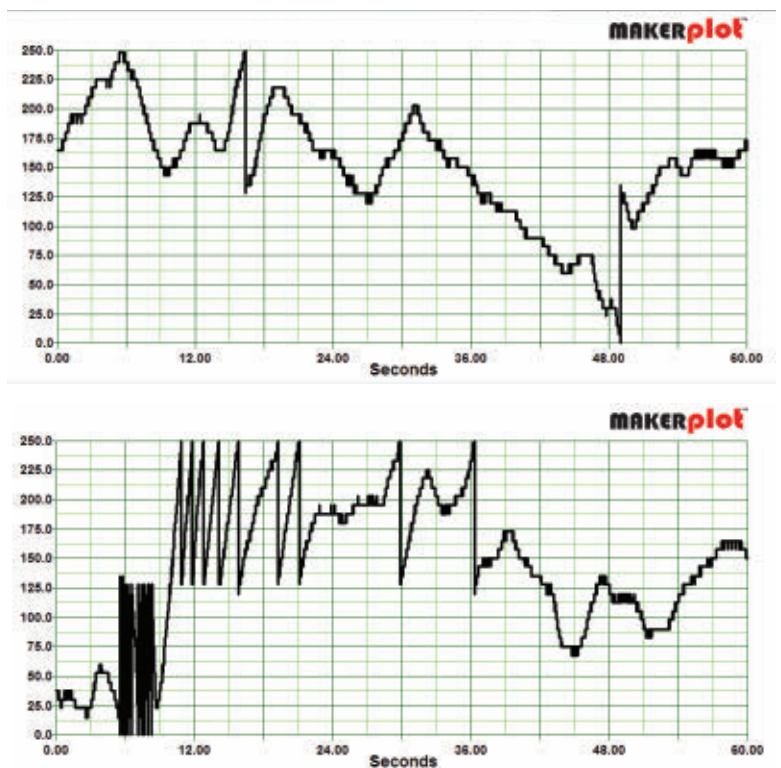


Figure 17. Recording of the eight-bit signal to drive the LED using MakerPlot, when the thermistor was touched and rapidly cooled, and then allowed to warm up. Note the rapid re-centering of the processed signal to the low side and then to the high side, to reach a more steady value.



TS-7970

Industrial High Performance i.MX6
Computer with Wireless Connectivity
and Dual GbEth

- 1 GHz Single or Quad Core Cortex A9 ARM CPU
- 2 GB DDR3 RAM (Quad) and 4 GB MLC eMMC Flash
- Bluetooth 4.0+EDR and WiFi 802.11BGN onboard radios

www.embeddedARM.com



ELECTRONET

USB Add USB to your next project--
It's easier than you might think!

USB-FIFO • USB-UART • USB/Microcontroller Boards
RFID Readers • Design/Manufacturing Services Available
Absolutely NO driver software development required!
www.dlpdesign.com

DLP Design

ALL ELECTRONICS CORPORATION
Electronic Parts and Supplies.
www.allelectronics.com
Free 96 page catalog 1-800-826-5432

For the ElectroNet
online, go to
www.nutsvolts.com
click **Electro-Net**

PCB, PCBA and More! **Myro** Low cost High Quality
www.myropcb.com
1-888-PCB-MYRO

Micro BGA SMT Adapters
0.5mm Pitch

 **Ironwood ELECTRONICS** 
www.ironwoodelectronics.com

superbrightleds.com
COMPONENT LEDs • LED BULBS • LED ACCENT LIGHTS



AGENT 330
TRACKED ROBOT KIT
\$329.99 @ ServoCity.com



Wanna read even more about electronics?
Check out our book selection at
store.nutsvolts.com

**Did You Know Preferred
Subscribers get access to all the
digital back issues of
Nuts & Volts for free?**

Call for details
1-877-525-2539

INVEST in your BOT!



 INVEST in **HITEC**

12115 Paine Street • Poway, CA 92064 • 858-748-6948 • www.hitecrd.com

MOBILE APP NOW AVAILABLE!

SERVO

Download NOW On Your
Favorite Mobile Device!

FOR ROBOT BUILDERS

iOS • ANDROID • KINDLE FIRE



www.servomagazine.com

**Did you know that Nuts & Volts
now has a weekly content
newsletter? *Want to get it?***

You have three ways to sign up:

- Visit us on Facebook and click on "Join My List."
- Using your cell phone, send "NVNEWSLETTER" as a text message to 22828.
- Visit the Nuts & Volts FAQs to sign up at nutsvolts.com/faqs

ROBOGAMES



Compete at RoboGames!

Last year, over 1000 builders from around the world brought over 800 robots to San Mateo, in the 11th annual international event. This year, we expect even more robots and engineers to compete. Be one! With 80 different events, there's a competition for everyone - combat, androids, sumo, soccer, Lego, art, micromouse, BEAM, or Tetsujin! More than half the events are autonomous. Even if you just come to watch, you'll be overwhelmed with the diversity.

Last year, RoboGames hosted teams with over 800 robots from Argentina, Australia, Austria, Brazil, Canada, China, Colombia, Czech Republic, Denmark, Germany, India, Indonesia, Iran, Japan, Korea, Mexico, Netherlands, Peru, Singapore, Slovenia, Sweden, Taiwan, UK, and the USA.

Be a RoboGames Sponsor!

RoboGames is the world's largest open robot competition - letting people of any age, gender, nationality, or affiliation compete. Sponsoring RoboGames not only helps more people to compete, but also gets your company unrivaled press coverage and visibility. The event has been covered by CNN, ESPN, Fox, CBS, ABC, NBC (live), EBS Korea, NHK Japan, BBC, and countless print and web companies. Your logo can be everywhere the cameras turn!

Rent a Booth!

Booth spaces are at the front of the venue, ensuring lots of traffic. With 3000-5000 people each day, your company will get amazing traffic!

World's Largest Robot Competition
-Guinness Book of Records

North America's Top Ten Geek Fests
-Wired Magazine

SportCenter's Top Ten
-ESPN SportsCenter

"If you are a robot enthusiast, I would definitely encourage you to attend the RoboGames... Take a plane, train, space elevator, but definitely go!"
-Servo Magazine

"Impossible to Imagine, Impossible to Forget!"
-Robot Magazine

Events:

Combat: 340 lbs, 220, 120, 60, 30, 3, & 1 lbs

Androids: Wrestling, Demonstration, Stair Climbing, The Eagle, Door Opening, The Toss, Basketball, Lift and Carry, Marathon, Obstacle Run, Penalty Kick, Dash, 3:3 Soccer, Weight Lifting

Open Events: Fire-Fighting, Robomagellan, Maze/MicroMouse, Walker Challenge, Biped Race, Robot Triathlon, Line Slalom, Ribbon Climber, Vex Open, Lego Challenge, Lego Open, Aibo Performer, Balancer Race, Best of Show, Bot Hockey

Sumo: 3kg - Auto & R/C, 500g, 100g, 25g, Humanoid

Robot Soccer: Biped 3:3 & 5:5, Mirobot 5:5 & 11:11

Junior League: Lego Challenge, Lego Open, Lego Magellan, Woots & Snarks, Handy Board Ball, BotsketBall, 500 g Sumo, 120 lb combat, Best of Show, Vex Open

Tetsujin (ExoSkeleton): Lifting, Walking, Carrying

Art Bots: Static, Kinetic, Bartending, Musical, Drawing

BEAM: Speeder, Photovore, RoboSapien Hacker



April 8-10, 2016

Pleasanton, CA

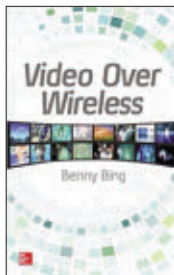
<http://RoboGames.Net>

GREAT FOR DIYers!

Video Over Wireless by Benny Bing

This practical volume offers expert guidance on delivering high quality video communications over wireless networks.

Wireless access has become the dominant medium for network connectivity due to the proliferation of smart-phones and tablets. This guide explains emerging technologies that will help overcome these challenges — all presented in a clear and concise format.

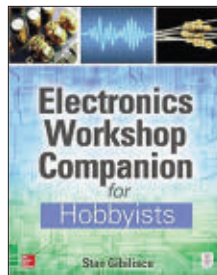


Reg. Price \$120.00 Sale Price \$ 101.95

Electronics Workshop Companion for Hobbyists by Stan Gibilisco

In this practical guide, electronics expert Stan Gibilisco shows you, step by step, how to set up a home workshop so you can invent, design, build, test, and repair electronic circuits and gadgets. Electronics

Workshop Companion for Hobbyists provides tips for constructing your workbench and stocking it with the tools, components, and test equipment you'll need. Clear illustrations and interesting do-it-yourself experiments are included throughout this hands-on resource. **\$25.00**

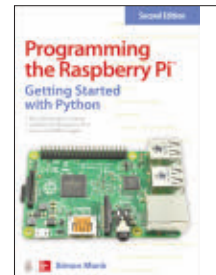


Programming the Raspberry Pi, Second Edition: Getting Started with Python

by Jack Simon

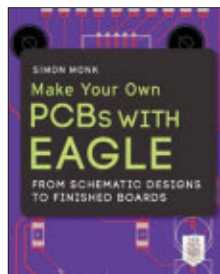
This practical book has been revised to fully cover the new Raspberry Pi 2, including upgrades to the Raspbian operating system. Discover how to configure hardware and software, write Python scripts, create user-friendly GUIs, and control external electronics. DIY projects include a hangman game, RGB LED controller, digital clock, and RasPiRobot complete with an ultrasonic rangefinder

\$15.00



Make Your Own PCBs with EAGLE by Eric Kleinert

Featuring detailed illustrations and step-by-step instructions, *Make Your Own PCBs with EAGLE* leads you through the process of designing a schematic and transforming it into a PCB layout. You'll then move on to fabrication via the generation of standard Gerber files for submission to a PCB manufacturing service. This practical guide offers an accessible, logical way to learn EAGLE and start producing PCBs as quickly as possible.



\$30.00

Programming the Intel Edison: Getting Started with Processing and Python by Donald Norris

Learn To Easily Create Robotic, IoT, and Wearable Electronic Gadgets!

Discover how to set up components, connect your PC or Mac, build Python applications, and use USB, Wi-Fi, and Bluetooth connections. Start-to-finish example projects include a motor controller, home temperature system, robotic car, and wearable hospital alert sensor.

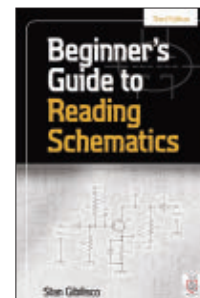
\$20.00



Beginner's Guide to Reading Schematics, 3E by Stan Gibilisco

Navigate the roadmaps of simple electronic circuits and complex systems with help from an experienced engineer. With all-new art and demo circuits you can build, this hands-on, illustrated guide explains how to understand and create high-precision electronics diagrams. Find out how to identify parts and connections, decipher element ratings, and apply diagram-based information in your own projects.

\$25.00



How to Diagnose and Fix Everything Electronic by Michael Jay Geier

Master the Art of Electronics Repair

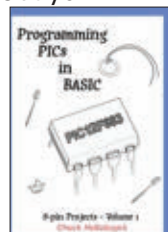
In this hands-on guide, a lifelong electronics repair guru shares his tested techniques and invaluable insights. *How to Diagnose and Fix Everything*

Electronic shows you how to repair and extend the life of all kinds of solid-state devices, from modern digital gadgetry to cherished analog products of yesteryear. **\$24.95**



Programming PICs in Basic by Chuck Hellebuyck

If you wanted to learn how to program microcontrollers, then you've found the right book! Microchip PIC microcontrollers are being designed into electronics throughout the world and none is more popular than the eight-pin version. Now the home hobbyist can create projects with these little microcontrollers using a low cost development tool called the CHIPAXE system and the Basic software language. Chuck Hellebuyck introduces how to use this development setup to build useful projects with an eight-pin PIC12F683 microcontroller. **\$14.95**



Programming Arduino Next Steps: Going Further with Sketches by Simon Monk

In this practical guide, electronics guru Simon Monk takes you under the hood of Arduino and reveals professional programming secrets. Also shows you how to use interrupts, manage memory, program for the Internet, maximize serial communications, perform digital signal processing, and much more. All of the 75+ example sketches featured in the book are available for download. **\$20.00**



Order online @ store.nutsvolts.com
Or CALL 1-800-783-4624 today!

EDUCATIONAL

Beginner's Guide Educational Combo!

Price Reduced
Now Only \$219.95

An Arduino Workshop

Are you puzzled about the Arduino but finding it difficult to get all the pieces in one place?

Joe Pardue
SmileyMicros.com

Puzzled by the Arduino?

Based on *Nuts & Volts* Smiley's Workshop, this set gives you all the pieces you need!

Educational Combo \$124.95

Arduino 101

arduinoclassroom.com/index.php/arduino-101

Arduino Classroom - learn computing and electronics

Home Arduino 101 Forum Blog Site Map Contacts

The free Internet virtual textbook: Arduino 101 at www.arduinoclassroom.com provides a sensible learning sequence that introduces computing and electronics with clear text and detailed hands-on labs with tested examples using the Arduino Projects Kit.

Available from Nuts&Volts for only \$44.99

CD-ROM SPECIAL

**Nuts & Volts
12 CD-ROMs
& Hat Special!**

That's 144 issues.
Complete with supporting
code and media files.

Free Shipping!

**Only \$249.95
or \$24.95 each.**

**The Nuts & Volts
Pocket Ref**

All the info you need at your fingertips!

This great little book is a concise all-purpose reference featuring hundreds of tables, maps, formulas, constants & conversions.

Only \$12.95

AND it still fits in your shirt pocket!

Visit <http://store.nutsvolts.com> or call (800) 783-4624

JumpStart Your ARM Development

There is a special place in my heart for the CCS C compiler. C compilers that target the PIC microcontroller are numerous. However, the CCS C compiler has a unique touch and feel. The flavor that this C compiler exhibits is directly coupled to its abundance of built-in functions that are specifically aimed at PICs. If your target microcontrollers are ARM-based, the same can be said for JumpStart C for Cortex-M. JumpStart C is an ARM Cortex-M C compiler environment that utilizes the ImageCraft JumpStart C API to enable rapid ARM application development. The JumpStart C API provides an abstraction layer that encapsulates the minute details associated with ARM microcontroller programming. JumpStart C for Cortex-M comes packaged within its own IDE (integrated development environment) and even includes a resident debugger.

Passing Muster

For JumpStart C for Cortex-M to be useful, it must be flexible. Many of the ARM development boards offer onboard programming/debugging via STMicroelectronics' ST-LINK/V2 or Segger's J-Link OB. JumpStart C is capable of supporting both onboard debugger/programmer solutions. However, if you design microcontroller based gadgets for a living, odds are you won't be offering your design to a customer by way of an off-the-shelf ARM development board. So, like CCS C for PICs, JumpStart C must also be able to interface to the professional tools

that support ARM microcontroller application development. In my ARM world, that professional tool comes in the form of a Segger J-Link PRO.

The Segger J-Link PRO captured in **Photo 1** is a very capable ARM development tool. The J-Link PRO is actually a refined version of the standard J-Link. The J-Link PRO adds an Ethernet interface and a pair of additional LEDs which act as status indicators. Choose your poison: Windows, Linux, or MAC. The J-Link PRO can operate with them all using either its Ethernet or USB interface. SWD (Serial Wire Debug) and JTAG interfaces are supported. I don't think you can throw an ARM microcontroller at the J-Link PRO that it doesn't know about.

According to **Screenshot 1**, the JumpStart C development environment embraces both the ST-LINK/V2 and J-Link PRO. There is only one way to insure that the radio button selections we are privy to in **Screenshot 1** will yield the required result.

ARMing a Serial Port

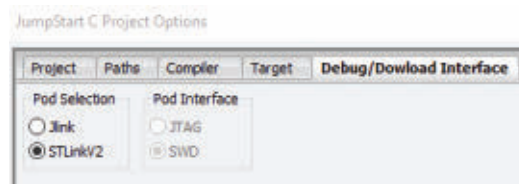
When it comes to serial ports, timing is everything. So, we have to first initialize our system clock. JumpStart C takes the pain out of clock configuration with the *SetSystemClock* function:

```
jsapi_clock.SetSystemClock(8,0,0,0,48);
```

The STM32F030R8T6 based development board you see in **Photo 2** has printed circuit board (PCB) pads for an external crystal, which are not loaded by default. That means (for now) our *SetSystemClock* function call will have to specify the system clock source as internal (HSI). This is signified by the 8 in our function call. The units are MHz, so to get a final PLL-generated 48 MHz system clock, we simply end our *SetSystemClock* function call with 48. The *SetSystemClock* function along with all of the other JumpStart C API is documented in the *Jumpstart API for STM32F0xx and STM32F4xx* document.

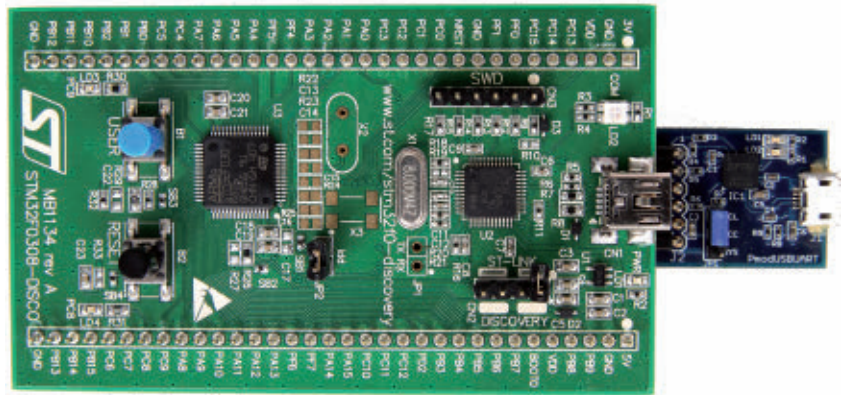


■ Photo 1. The J-Link PRO is a versatile microcontroller development tool. In addition to ARM, the J-Link PRO supports application development with Microchip's PIC32 and Renesas.



■ Screenshot 1. All we have to do is make sure that the correct J-Link and ST-LINK/V2 drivers are installed. JumpStart C can take it from there.

■ **Photo 2.** This is a stock STM32F030R8T6 Discovery evaluation platform. It has been modified to accept the 20-pin J-Link programmer cable. A Digilent PmodUSBUSART module has been added at the opposite end of the perfboard that comes with the evaluation board. The main board plugs into the perfboard.



To keep you from straining your neck going back and forth between the API specification and your application code, the JumpStart IDE automatically displays helper API call syntax as you enter the API call in your source code. While we are talking clocks, let's go ahead and set up a *SysTick* timer:

```
jsapi_cortex_core.SysTick_Timer(SYSTICK_MILLI  
SECOND);
```

Every millisecond, an interrupt will be generated and the JumpStart API can (and will) use this interrupt to keep time. This will enable us to use the JumpStart API delay functions. We can now turn our attention to the configuration of the USART. The STM32F030R8T6 supports a pair of USARTs. We will activate USART2:

```
usart2.SetPins(&porta, 2, 1, &porta, 3, 1);
```

The transmit pin for USART2 is located on PORTA, pin 2 (PA2). The receive pin is defined as PORTA, pin 3 (PA3). The Alternate Function Register for both transmit and receive alternate functions is AF1, which is denoted by the pair of 1s in the *SetPins* function argument.

Now that our USART pins have been defined and activated, we can set the USART's baud rate, number of data bits, number of stop bits, and flow control:

```
usart2.MakeUSART(9600, 8, 1, 0);
```

Let's combine our clock and USART code into a single function called *Setup*:

```
static void Setup(void)
{
    jsapi_clock.SetSystemClock(8,0,0,0,48);
    //jsapi_clock.SetSystemClock(0,0,8,0,48);
    jsapi_cortex_core.SysTick_Timer
    (SYSTICK_MILLISECOND);
    usart2.SetPins(&porta, 2, 1, &porta, 3, 1);
    usart2.MakeUSART(9600, 8, 1, 0);
}
```

The commented *SetSystemClock* entry is the code we can use to configure our system clock as HSE with an external 8 MHz crystal.

At this point, we can bang single characters through our new USART at will. However, having the ability to employ the features of the *printf* command allows us to

easily send formatted messages via our USART. To get *printf* into the game, we need to write a local or top level *putchar* function that will override the *putchar* function that is contained within the *stdio* library:

```
int putchar(unsigned char ch)
{
    if (ch == '\n')
        usart2.putchar('\r');
    usart2.putchar(ch);
    return ch;
}
```

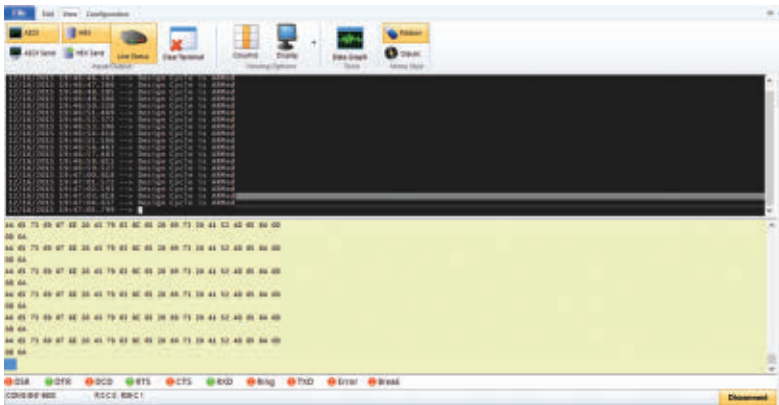
Our top level *putchar* function utilizes the JumpStart API *putchar* call. This allows us to use the *printf* function in our code. We should now be able to write some serial transmission code that can use *printf* formatting. We also configured a *SysTick* timer that should enable us to code in a delay between transmissions. Here's the resultant source:

```
int main(void)
{
    Setup();

    while (1)
    {
        printf("Design Cycle is ARMed\r\n");
        DelayTenth(5);
    }
    return 0;
}
```

If all works as designed, our little app should transmit the Design Cycle message every half second. Before we can see any results, we need to install some serial port hardware. The STM32F030R8T6 evaluation board captured in **Photo 2** has been modified to accept the J-Link PRO 20-pin programming cable. On the opposite end of the STM32F030R8T6 board, a Digilent PmodUSBUSART module has been wired in. It's a simple connection.

The Pmod's TX pin is connected to the STM32F030R8T6's PA3 RX pin. PA2 — the STM32F030R8T6's TX pin — is connected to the Pmod's RX pin. No power connections other than GND are necessary as the Pmod is powered from its USB portal. I used a six-pin, right angle, 0.1 inch pitch female socket to



■ **Screenshot 2.** This screen capture leaves no doubt that our application of the JumpStart API is working as designed. Don't get too excited about the time stamps. These times are relative to the terminal emulator, not our STM32F030R8T6 *SysTick* timer.

mount the Pmod. The USART program compiled without error. A debug session was kicked off with the J-Link PRO connected to the STM32F030R8T6. I fired up the CCS SIOUW terminal emulator and configured it for 9600 baud. Once the emulator came up, I was presented with a pair of green LED status indicators on the J-Link PRO and a blinking transmit LED on the Pmod. **Screenshot 2** confirms our success with the ARM USART.

I SPI

Now that we have established a working USART portal that we can use as a debug tool, let's bring up another important communications port we all know as the SPI portal. We shall begin by consulting the STM32F030R8T6 datasheet and laying out the SPI MOSI, MISO, SCLK, and CS pins:

```
spil.SetPins(
    &porta, 5, 0,          // SCK
    &porta, 7, 0,          // MOSI
    &porta, 6, 0,          // MISO
    &portb, 6,             // CS
    true                  // active LOW
);
spil.MakeSPI(8, 0, 400000);
```

The STM32F030R8T6 datasheet tells us that the SPI1 pins are located on PORTA in AF0 (Alternate Function 0). The SPI CS pin is not datasheet assigned. So, PB6 is just as

good as any here. The CS pin is configured as active low. I have plans for this SPI portal. So, for now, we will configure it as an eight-bit portal running in SPI mode 0 at 400 kHz. SPI mode 0 equates to a clocked data stream with phase low (CPHA = 0) and polarity low (CPOL = 0). Now that we have defined, configured, and activated our SPI portal, the JumpStart API supports it with a load of SPI data transfer calls.

For instance, we can send eight bits of data to an SPI slave with this JumpStart API call:

```
bite = spil.Write(data);
```

As you can see, the *spi1.Write* call also returns a byte of information from the slave. If the application calls for 16 bits per transfer, all we have to do is alter our *spi1.MadeSPI* call like this:

```
spil.MakeSPI(16, 0, 400000);
```

Controlling the SPI chip select (CS) line is also covered by the JumpStart API:

```
spil.ChipSelect();          //CS = low
spil.ChipDeselect();        //CS = high
```

Remember that I stated earlier that I had a "plan" for our SPI portal?

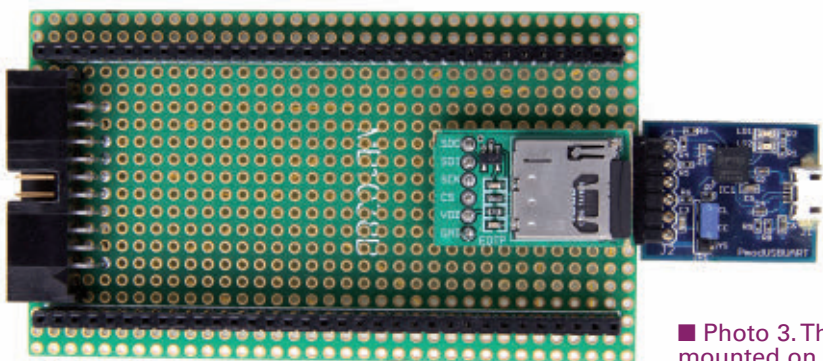
microSD ARMed

The reconnaissance data captured in **Photo 3** reveals that a microSD card socket and associated components have been added to our STM32F030R8T6 ARM complex. The microSD card socket assembly is mounted on top of the six-pin Pmod access socket. Everything is wired in

using point-to-point techniques. **Schematic 1** outlines the attachment of the 20-pin JTAG programming/debugging header.

The microSD circuitry is outlined graphically in **Schematic 2**. The tristate buffer in the design provides output buffering, as well as isolation from the STM32F030R8T6's SPI MISO line.

The microSD's MISO and CS signals should be pulled up, and we have done



■ **Photo 3.** The microSD card socket and buffer circuitry are mounted on top of the Pmod six-pin socket.

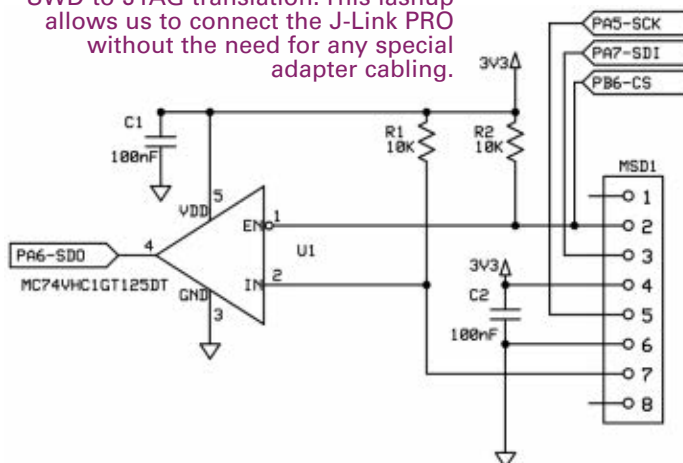
JumpStart API
JumpStartMicroBox
ImageCraft
www.imagecraft.com

CCS C Compiler
CCS, Inc.
www.ccsinfo.com

J-Link PRO
Segger
www.segger.com

Digilent
PmodUSBUART
www.digilentinc.com

■ Schematic 1. This is the classic SWD-to-JTAG translation. This lashup allows us to connect the J-Link PRO without the need for any special adapter cabling.



just that in our circuit design.

The hardware is fairly straightforward. Thanks to the JumpStart API, the firmware is too. The JumpStart API package includes a number of example projects including a microSD project that implements the open source FatFs. FatFs is a generic FAT file system module. The example FatFs core files in my JumpStart API package were down a level. So, I downloaded the current set of FatFs core files and replaced them in the JumpStart API example code.

The replacement of the existing FatFs core files necessitated changes to the JumpStart API's fatfs_diskio_sdcard_spi.c file. The changes involved syncing the variable type declarations in the fatfs_diskio_sdcard_spi.c file with those in the latest FatFs integer.h file. Don't worry; I've included my updated file in a download package for you at the article link.

Let's see if we can get this microSD thingie to work. Since reading and writing the SPI portal is key, we'll need to package the JumpStart API SPI write call into a function:

```
unsigned char spi_txx(unsigned char data)
{
    return spil.Write(data);
}
```

The fatfs_diskio_sdcard_spi.c file is a combination of low level SPI routines coupled with glue functions that tie the low level SPI functions to the upper level FatFs application calls. The JumpStart API example is well written, but it is wordy. The best way to get our arms around it is to work using the FatFs Device Control Interface as our reference point. The Device Control Interface is predefined in the FatFs specification. It consists of six calls:

disk_status — Get device status
disk_initialize — Initialize device
disk_read — Read sector(s)
disk_write — Write sector(s)
disk_ioctl — Control device dependent features
get_fattime — Get current time

Let's take a look at the disk_status interface call:

```
typedef BYTE DSTATUS;
```

```
typedef struct hwif
```

```
{
    int initialized;
    int sectors;
    int erase_sectors;
    int capabilities;
} hwif_t;

hwif_t hw = {0,0,0,0};

DSTATUS disk_status(BYTE drv)
{
    if (hw.initialized)
        return 0;

    return STA_NOINIT;
}
```

I've taken the liberty to assemble associated items found in supporting files. The structure variable *hw.initialized* is influenced by lower level calls and operations. All that FatFs cares about the variable is that it has a valid value when FatFs comes to call for disk status at the application level.

The *disk_initialize* interface depends on the outcome of the *hwif_init* function. The final state of the *hwif_init* function is dependent on the results of the *sd_init* function:

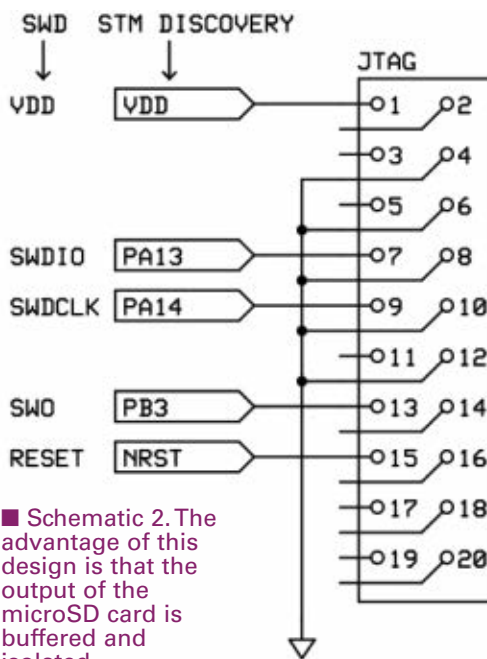
```
int hwif_init(hwif_t* hw)
{
    int tries = 10;

    if (hw->initialized)
        return 0;

    while (tries--)
    {
        if (sd_init(hw) == 0)
            break;
    }

    if (tries == -1)
        return -1;

    DSTATUS disk_initialize
    (BYTE drv)
    {
        if (hwif_init(&hw) == 0)
            return 0;
    }
}
```

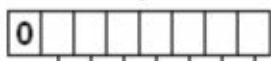


■ Schematic 2. The advantage of this design is that the output of the microSD card is buffered and isolated.

ImageCraft is giving away three JumpStart MicroBoxes to randomly selected *Nuts & Volts* readers. Just send an email on March 31, 2016 or earlier to info@imagecraft.com with the words NUTSNVOLTS-GIVEAWAY in the subject field to qualify.

■ Screenshot 3. The R1 response byte value we are looking for after the 74+ clocks is *In Idle State*.

R1 Response



In Idle State

Erase Reset

Illegal Command

Command CRC Error

Erase Sequence Error

Address Error

Parameter Error

```
return STA_NOINIT;
}
```

The *sd_init* function is lengthy. So, let's assimilate it in the way we would go about eating an elephant. The SD card specification says to begin by sending 74 or more clocks to the microSD card with the CS signal logically low:

```
printf("cmd0 - reset.. ");
spi1.ChipSelect();
spi1.ChipDeselect();
/* 74+ clocks with CS high */
for (int i = 0; i < 12; i++)
    spi1.Write(0xFF);
```

As you can see, the JumpStart API *spi1* calls make this a simple matter. The next step involves reading the microSD card and interpreting the response, which is called R1 in the spec:

```
/* reset */
spi1.ChipSelect();
sd_cmd(0,0);
r = sd_get_r1();

static BYTE sd_get_r1(void)
{
    int tries = 1000;
    BYTE r;

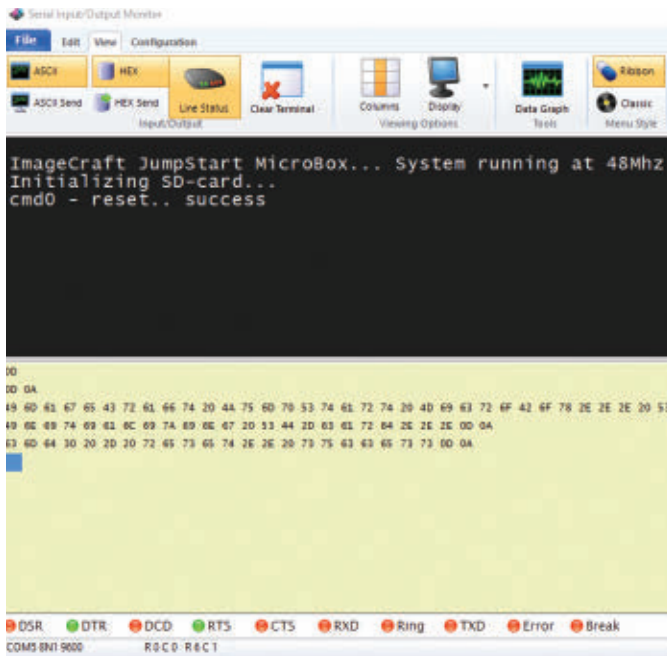
    while (tries-- != 0) {
        r = spi1_txx((BYTE)0xff);
        if ((r & 0x80) == (BYTE)0)
            return r;
    }
    return (BYTE) 0xff;
}
```

If you follow through the *sd_init* code in your download package, you will see that we ultimately want to see an R1 response value of 0x01. The R1 bit scheme is laid out in **Screenshot 3**.

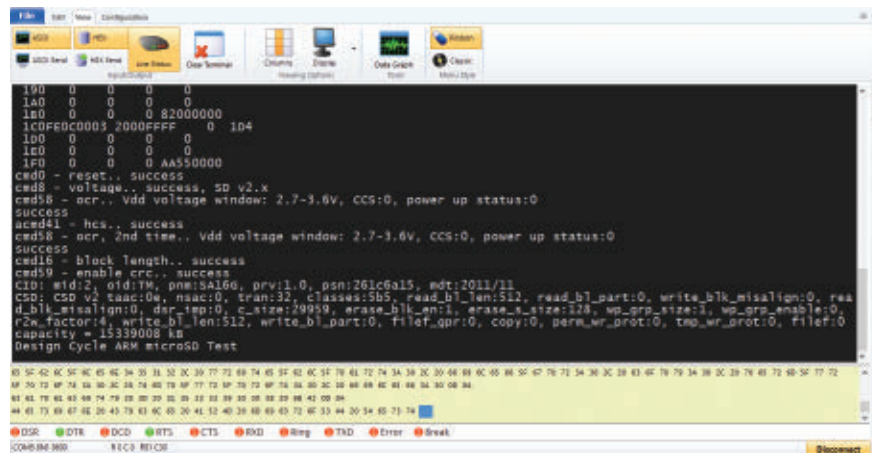
We're off to a good start. **Screenshot 4** informs us that the microSD card has come to life. At this point, we can issue other microSD card commands to gather configuration information and even read the card. The demo contains various commands which are embedded into the *sd_init* function. However, the *sd_init* function does not issue any FatFs commands to actually read a file. So, I have added some file access code to the main.c code:

```
FATFS FatFs;
FIL fil;
char line[32];
FRESULT fr;

f_mount(&FatFs,"",0);
fr = f_open(&fil,"message.txt",FA_READ);
if(fr)
    return (int)fr;
while(f_gets(line,sizeof line,&fil))
```



■ Screenshot 4. So far, so good. Now that we have the microSD card's attention, we can issue all sorts of commands to gather configuration information.



■ Screenshot 5. The command execution follows the source code found in the main.c file (in the examples.JumpStartMicroBox/STM32F030/SDcard folder) and fatfs_diskio_sdcard_spi.c file, which is located in the examples.JumpStartMicroBox/FatFs folder.

■ **Photo 4.** In this shot, you can see the additional capacitors, resistors, and crystals that are absent in **Photo 2**. The steps needed to add these components are outlined in the STM32F0 Discovery user manual.

```
printf(line);
fclose(&fil);
```

The text file message.txt contains a single string of characters. The FatFs code snippet mounts the volume, opens the file in read only mode, and retrieves the string into the *line* array. The contents of the array are then sent to the serial port. Once the file contents have been read and displayed, the text file is closed.

The FatFs top level commands make it look easy. Study the `fatfs_diskio_sdcard_spi.c` code and you will see that there are many supporting functions that perform the actual reading and writing of the microSD card media.

Does It Work?

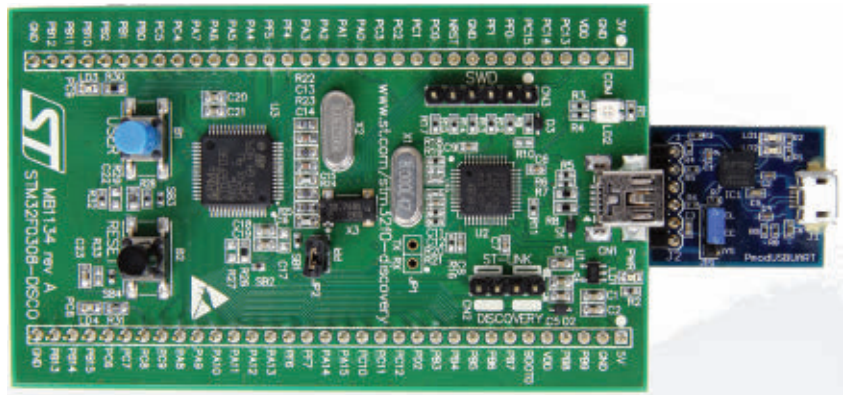
Well, we know that the USART is operational. We also know that we can successfully bring up the microSD card. Before we turn the rest of the `sd_init` code loose, let's fiddle with the hardware again. Recall that I coded (and commented out) a line to drive the STM32F030R8T6 from an external 8 MHz crystal:

```
jsapi_clock.SetSystemClock
(8,0,0,48);
//HSI internal clock
//jsapi_clock.SetSystemClock
(0,0,8,48);
//HSE 8MHz crystal clock
```

Let's remove the comment characters from the HSE code, comment the HSI clock code, and run the final test using the HSE clock instead of the internal HSI clock.

The command execution follows the source code found in the `main.c` file (in the `examples.JumpStart`

`MicroBox/STM32F030/SDcard` folder) and `fatfs_diskio_sdcard_spi.c` file, which is located in the `examples.JumpStartMicroBox /FatFs` folder. The last thing we should see is the contents of the `message.txt` file. I let the microSD card code run all the way through. It produced the information you see captured by the



SIOW program in **Screenshot 5**. I loaded the microSD card socket with a 16 GB SDHC microSD card. All of the data on this card we could extract is listed in the **Screenshot 5** report.

I'll leave you with **Photo 4**, which reveals an STM32F030R8T6 Discovery Board with the addition of an 8 MHz crystal, its associated components, and a 32.768 kHz clock crystal with its set of capacitors. You can now add the JumpStart API and STM32F030R8T6 to your Design Cycle. **NV**

IMAGECRAFT

EMBEDDED SYSTEMS

JumpStart C Tools for AVR & Cortex-M

with JumpStart API and IDE / debugger, start Cortex-M C programming in minutes, not days.

Starts at \$99
Beyond Arduino - when you're ready to get serious



wifi2go
Wifi2go IoT design module

OEM production ready, volume pricing, STM32 MCU + TI CC3100 + JumpStart API



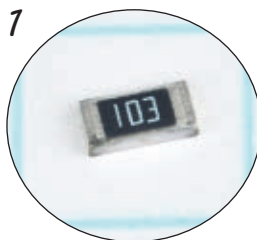
Find us on

<http://imagecraft.com> • info@imagecraft.com

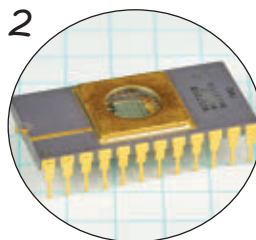
Name that Part!

Try this photo quiz and see how many parts you can identify. Some parts date back to the 1950s and earlier, while others can be found at your local RadioShack. For scale, the blue background grid contains 1/4 inch squares. Keep in the mind the photos have been sized to fit the layout. The correct answers can be found on page 65. Good luck!

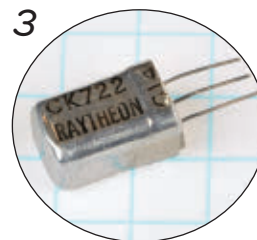
Scoring: 0-7 = Novice 8-15 = Good
16-18 = Expert



- a. 10K SMD Resistor
- b. 1K SMD Resistor
- c. 103 ohm Resistor



- a. 2716 EPROM
- b. 8080 CPU IC
- c. 6502 CPU IC



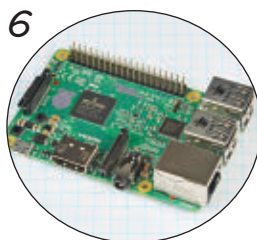
- a. NPN Transistor
- b. CK722 Triac
- c. PNP Transistor



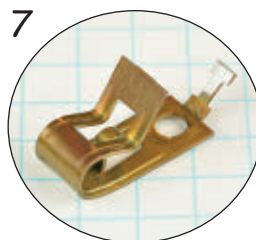
- a. Quad Nand IC
- b. Quad Op-amp IC
- c. 324 Multiplexer IC



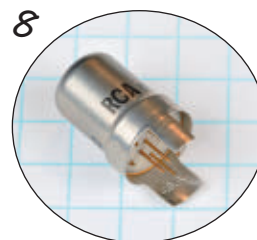
- a. 913 1" CRT Tube
- b. X-ray Tube
- c. 913 Power Tetrode



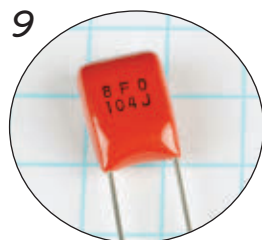
- a. BeagleBone Board
- b. Arduino Uno Board
- c. Raspberry Pi Board



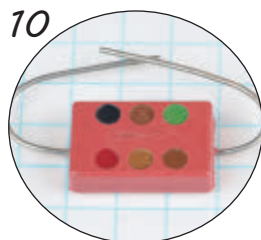
- a. Fleming Clip
- b. Fahnestock Clip
- c. Farnsworth Clip



- a. Thyristor
- b. Nuvistor
- c. Varistor



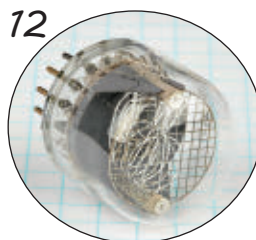
- a. 104 pF Capacitor
- b. 0.1 μ F Capacitor
- c. 104 μ F Capacitor



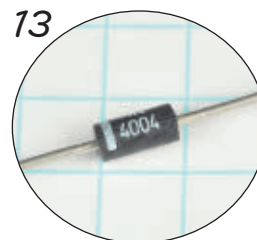
- a. Mica Capacitor
- b. Electrolytic Capacitor
- c. Tantalum Capacitor



- a. USB Type A
- b. USB Type B
- c. USB Type C



- a. Nixie Tube
- b. Numitron Tube
- c. Dekatron Tube



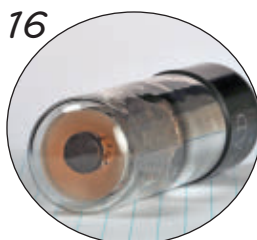
- a. 40 amp Diode
- b. Signal Diode
- c. 400V Diode



- a. Voltage Regulator
- b. Full Wave Rectifier
- c. Selenium Rectifier



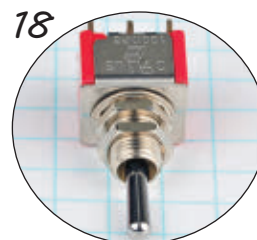
- a. 555 Timer IC
- b. 555 Buffer IC
- c. 555 Inverter IC



- a. Rectifier Tube
- b. 1629 X-ray Tube
- c. Magic Eye Tube



- a. SPST Relay
- b. SPDT Relay
- c. DPDT Relay



- a. OFF-ON-OFF
- b. ON-OFF
- c. ON-OFF-ON

Presented by David Goodsell

Several of my electronics friends contributed parts and reviews to this quiz, and I would like to thank them:
A. Abbott, A. Borack, B. Douglass, R. Hepperle, and R. Heryford.

Impedance Matching

Impedance matching is not mysterious, it is really just about controlling reflections.

The previous column on transmission lines (a.k.a., feed lines) and standing wave ratio (or SWR) explained how RF energy travels in the line and what happens when it encounters a load or different type of line. The resulting reflections create standing waves of voltage and current, just as water waves do when they encounter an obstruction. As mentioned last time, the RF energy bouncing around in the line can create problems with digital signals and can result in extra losses.

This month, we'll take a look at several techniques for dealing with the reflections and how the same techniques are used in circuit design. This is called *impedance matching*.

Review of Reflections

No matter what kind of wave we are talking about — electromagnetic (EM), acoustic, or ripples on a rope — whenever the wave encounters a change or *discontinuity* in the impedance of whatever medium it's flowing in or on, the wave's components have to change as well. For example, if we connect a 50Ω RG-58 cable to a 75Ω RG-6, the electric and magnetic field components of the wave change at the junction of the two cables. The change in impedance is called a *mismatch*.

To create the abrupt change, the wave splits into two new waves traveling in opposite directions but with the same total amount of energy. One wave travels into the new medium (the *forward* wave) and the other wave goes backward in the old medium (the *reflected* wave). The larger the discontinuity, the more energy remains in the reflected wave.

If the new medium's impedance is zero (a short circuit) or infinite (an open circuit), no energy is transferred into the new medium; all of the energy is reflected and the SWR is infinite. The wave is reflected again at the source and heads back in the original direction. The split-and-reflect process occurs at the discontinuity on every round trip. As the reflected wave travels back and forth, some of that energy is lost through resistance (for an EM wave) or friction (for a mechanical wave) along the way. (The video "Similarities in Wave Behavior" referenced in the previous column at www.youtube.com/watch?v=DovunOx1Y1k shows this process clearly.)

Imagine an RF signal from a transmitter in a piece of coaxial feed line connected to an antenna. Now, break up that RF signal into a sequence of packets, like boxcars on a train. **Figure 1** shows one such packet as it bounces back and forth, getting smaller and smaller as it transfers power to the antenna, and loses power from losses in the cable.

If the packet has a very short duration and you looked at it on an oscilloscope, you would see it coming and going with its shape determined by the various discontinuities in the feed line. In fact, that is exactly what a *time-domain reflectometer (TDR)* does: Generate a sharp pulse in the line and display it. The shape of the returned pulse and the time it takes to make the round trip show just where the discontinuity is and its impedance.

To an RF engineer, the reflections dissipate signals as heat in the feed line, create excess voltages and currents, and alter impedances. To a digital designer, the reflections on a printed circuit board (PCB) signal trace can distort rising and falling edges of digital pulses, creating false clocks, delays, and erroneous values. As clock speeds rise and transition times fall, eliminating these reflections can become quite important to reliable operation.

There are several ways of eliminating the reflections.

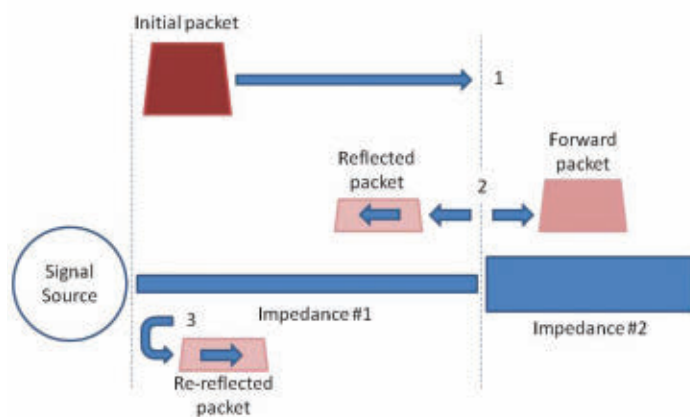


FIGURE 1. A packet of RF energy from the source travels to the end of the transmission line (1) with impedance #1 where it encounters transmission line with impedance #2. The packet splits (2) into a forward packet continuing on and a reflected packet headed back toward the source. At the source, the reflected packet is re-reflected (3) and the process repeats until all of the packet energy has been transferred to the second line or dissipated as heat.

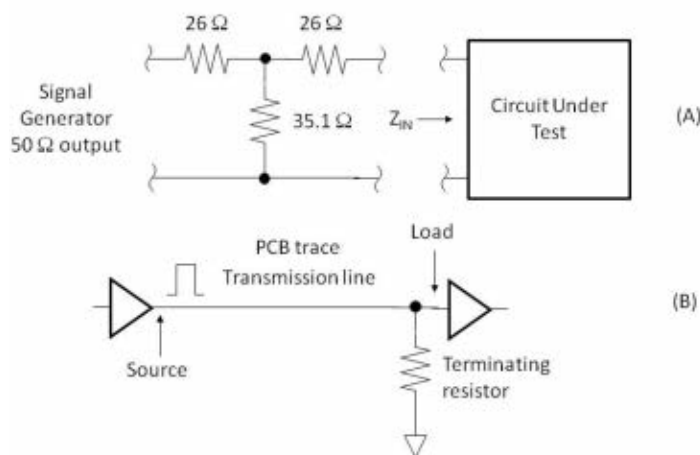


FIGURE 2. At A, a 10 dB attenuator insures that the signal generator output load is between 43.9Ω and 51.1Ω — no matter what load the CUT presents. This is useful in stabilizing the generator output as the circuit is adjusted. At B, a terminating resistor is used to make sure no reflections are generated when the pulse traveling along the PCB trace transmission line encounters the logic circuit's input.

Figure 2A shows a resistive T attenuator used to match impedances. The goal is to make the load presented to the signal generator close to 50Ω, no matter what value Z_{IN} of the Circuit Under Test (CUT) may be. In the process, however, the attenuator knocks the input signal power down by a factor of 10 (10 dB)! This only works if you have power to spare from the source.

If $Z_{IN} = \text{infinity}$ (an open circuit), then the generator “sees” the 26Ω and 35.1Ω resistors in series for 51.1Ω, and an SWR of $51.1/50 = 1.022$. If $Z_{IN} = 0$ (short circuit), the input impedance is $26 + 35.1 // 26 = 43.9\Omega$ for an SWR of $50/43.9 = 1.14$ ($//$ means “in parallel with”). Any other impedance will present a load between 43.9Ω and 51.1Ω, so the generator’s output will not be greatly affected by Z_{IN} .

Many test setups specify attenuators at the output of signal generators. It’s not because the generators are putting out too much signal; it is to present a stable impedance to the generator to avoid having to constantly readjust for a constant output level as a load changes.

Digital system designers can use resistors to *terminate* a PCB trace carrying high speed signals as shown in Figure 2B. The resistor is connected to common at an IC input. The termination makes sure no reflections are generated to travel back and forth on the trace. Each logic family has techniques for controlling reflections on signal lines. Check the datasheets for information on terminations and working with high speed signals.

Mechanical Impedance Matching

A less wasteful method of getting rid of the reflections is to transform the mismatched impedance to a different value that doesn’t cause reflections at all. The following analogy may help. A mechanical analog to electrical impedance — the ratio of voltage to current — is the ratio of torque (how hard you push) to rotational speed (how

Some work at a single frequency, and some over a wide range of frequencies. Some use electronic components, and some can even be constructed from more transmission line. Let’s meet a few of the more common tricks of the trade for impedance matching.

Resistive Impedance Matching

One of the simplest ways of getting rid of the reflections is to simply dissipate them as heat with resistors. The result is that the source of power is happy with its load, the output circuit is happy with the power delivered, and both are relatively isolated from the effects of the other. This isn’t impedance matching as much as it is “impedance hiding,” but the net result is the same: No reflections.

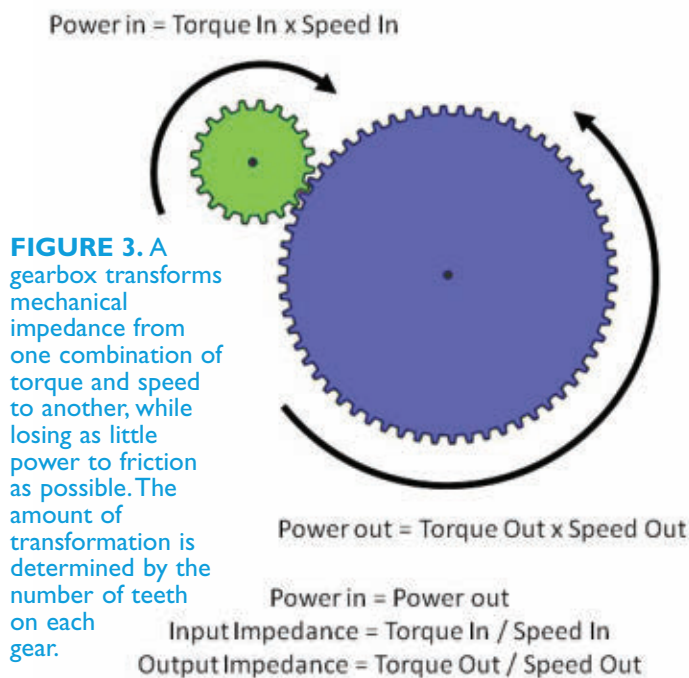
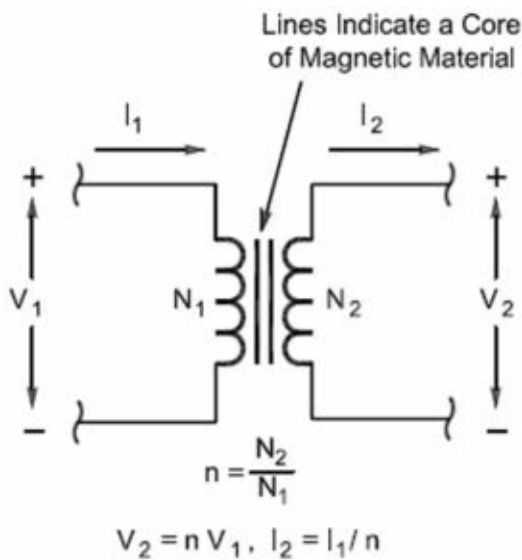


FIGURE 3. A gearbox transforms mechanical impedance from one combination of torque and speed to another, while losing as little power to friction as possible. The amount of transformation is determined by the number of teeth on each gear.

Microstrip Transmission Lines

Every signal-carrying PCB trace is a transmission line. For high speed signals, treating the traces as microstrip transmission lines helps avoid erratic operation from reflections. Microstrip is formed when a trace runs over or between ground planes — a common situation on multi-layer boards. You can learn more about microstrip at www.microwaves101.com/encyclopedias/microstrip and the Analog Devices tutorial “Dealing with High Speed Logic” (www.analog.com/media/en/training-seminars/tutorials/MT-097.pdf) is a good introduction.



$$Z_1 = V_1 / I_1 \quad Z_2 = V_2 / I_2 \quad Z_2 / Z_1 = n^2 \quad n = \sqrt{Z_2 / Z_1}$$

FIGURE 4. An electrical transformer transfers power with one combination of voltage and current (electrical impedance) through its core to another combination of voltage and current in the secondary circuit. The amount of transformation is determined by the ratio of turns on the transformer core's secondary winding to the number of turns on the primary winding.

fast the driven shaft turns). A load with high mechanical impedance takes a lot of torque to turn.

One mechanical impedance matching device we are all familiar with is the gearbox (or transmission) in **Figure 3**. The job of a car's transmission is to transfer power to the wheels which are turning at a wide range of speed, while keeping the engine happy near its optimum speed. It does so by converting one combination of torque and speed (impedance) to another, while losing as little power as possible to friction.

The situation is very similar to an RF transmitter designed for a load impedance of 50Ω but which can be connected to an antenna system presenting all sorts of different impedances. The electrical gearbox used here is known as an antenna tuner or transmatch.

One combination of voltage and current go in, and another combination of voltage and current come out, while heating up the matching unit with losses as little as possible. There are lots of different electronic components and circuits that can act as electrical gearboxes. Let's meet a few of those.

Transformers

Resistive matching is not very useful if the goal is to deliver power efficiently, such as via the power grid. Instead, we use a transformer to change the high voltages and low currents of power distribution lines to the low

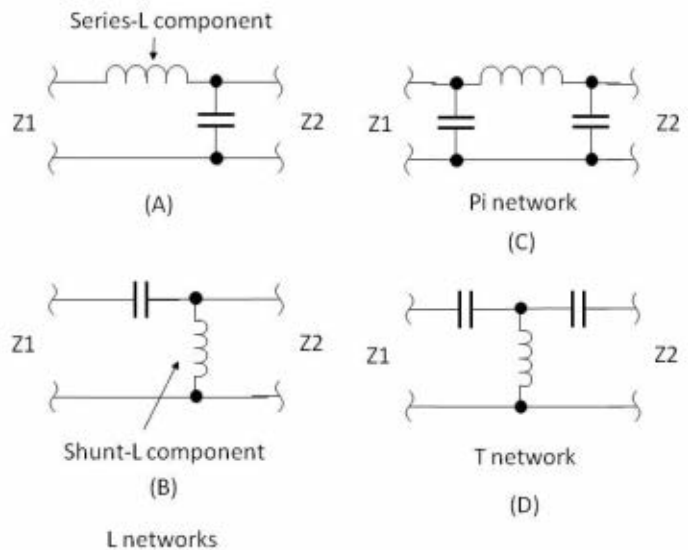


FIGURE 5. Four types of L-C impedance matching networks. Each type of network is named for the letter it resembles. A component in the signal path is called a series component. A component connected from the signal path to common or ground is called a shunt component. The circuit at A is a series-L shunt-C network, for example. The pi and T networks can each be viewed as back-to-back L networks as described in the text.

voltages and higher currents in homes and businesses. Power is transferred as magnetic energy from the transformer's primary to the secondary circuits through the transformer's core. This is called a flux-coupled transformer.

Figure 4 shows the basic math for how a transformer converts impedances. Let's say you have 115 VAC at the AC outlet and want 12 VAC for your project. Your step-down transformer would need a turns ratio of $n = 115 / 12 = 9.6$ from the secondary to the primary. If your project drew 1 A of current, that 12Ω impedance would be transformed to a load of $(12 \text{ V} / 1 \text{ A}) \times n^2 = 12 \times 92.2 = 1106\Omega$ to the AC line.

You can find impedance matching transformers in many different types of electronic circuits. A very common use is to match the output of a transistor audio amplifier (which may have an output impedance of several hundred ohms of resistance) into the 4, 8, or 32Ω of speakers or headphones.

In ham radio, it's not unusual for antenna systems to have impedances of 100Ω to $1,000\Omega$, so transformers are available with several selectable turns ratios to transform impedances by ratios of 2:1, 4:1, and 9:1. The result is an impedance that is close enough to 50Ω , so that either the transmitter is happy or only requires a little bit of impedance adjustment.

While power transformers for use at 50 Hz or 60 Hz have laminated steel or iron cores, transformers for RF use

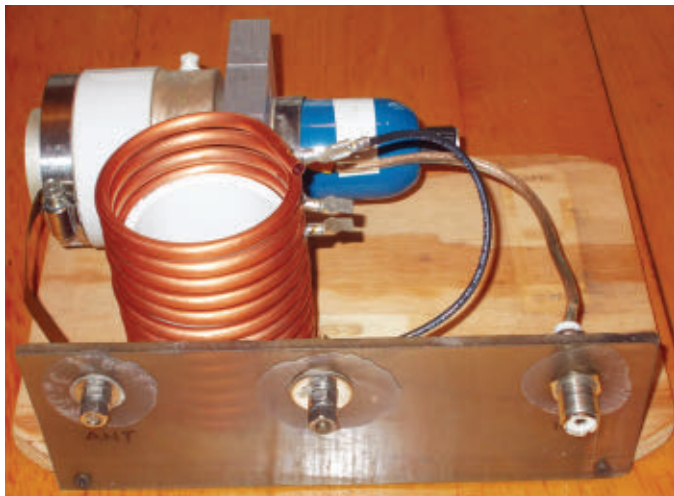


FIGURE 6. This L network is an example of the series-C shunt-L network in **Figure 5B**. Made from a heavy-duty coil and vacuum variable capacitor, it can easily handle 1 kW of RF.

powdered iron or ferrite cores. If carefully made, RF transformers can be used over a wide bandwidth of 10:1, such as from 3 MHz to 30 MHz or from 150 MHz to 1,500 MHz. RF transformers are available for receiving and small-signal applications of a few milliwatts to high power transmitting units.

L-C Impedance Matching

An alternative to transformers are the L-C circuits or networks shown in **Figure 5**. These networks are composed entirely of reactive components (inductance and capacitance). The combinations of reactances transform voltages and currents at one side of the network to a different combination at the other by sharing stored energy as circulating voltages and currents. This electrical juggling act only works when the reactances have the

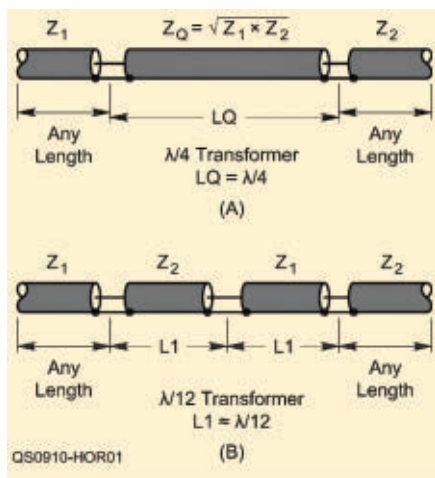


FIGURE 7. The quarter-wave transformer (or Q section) at A uses a quarter wavelength section of transmission line to match Z_1 and Z_2 by setting up reflections with the right phase to cancel all reflections from Z_2 . At B, two $\lambda/12$ wave sections match Z_1 to Z_2 without requiring a special section of transmission line.

(Graphic courtesy of the American Radio Relay League)

Online Network Calculator

There are many network calculator programs, such as the Microwaves 101 (www.microwaves101.com) and RF Café (www.rfcafe.com) websites. My favorite, though, is a web page by John Wetherell that calculates L and C values for 16 different networks (home.sandiego.edu/~ekim/e194rfs01/jwmatcher/matcher2.html). Enter the source and load impedance, then look at all of the choices for doing the job.

right values, and that only occurs at one frequency since reactance depends on frequency.

While four types of circuits are shown in the **figure**, there are many variations of each, and even other different circuits. Each of the networks is named for the letter which the arrangement of components resembles: L, pi (π), or T. Any of the Ls in the figure could be replaced by a C and vice versa, with the resulting circuit being able to transform different combinations of impedance.

The L network is very simple and works well for matching values of impedance that aren't too different. (The more different Z_1 and Z_2 are, the higher the amplitudes of the circulating voltages and currents, and the higher the losses in the circuit components.) Each variation of the L network is capable of matching half of the possible combinations of input and output impedance. The remaining combinations can be matched by turning the network around. There is no dedicated input or output!

A calculator program (see the **sidebar**) determines values of L and C that create the impedance match. Out of the four L networks with one L and one C, two of the circuits will "work." (For the other two circuits, no values of L and C are possible.) Of the two possible circuits, you can select the values of L and C that are most practical, the least expensive, or whatever is most suitable to your needs.

Figure 6 shows a high power L network that can handle more than a kilowatt of RF power at 3.55 MHz. Other L networks might handle only milliwatts of power.

The pi and T networks act like two L networks connected back-to-back. For example, if the series inductor of the pi network is replaced with a pair of inductors — each having half the original value — it is easy to see the input and output L networks. Similarly for the T network, the shunt inductor can be replaced with two double-value inductors in parallel, creating two back-to-back L networks. Why go to all that trouble when a simpler L network will do the job?

Remember that as the input and output impedances get farther apart, the circulating voltages and currents get larger. Another change is that the bandwidth over which the impedances can be considered matched shrinks. As the impedance ratio (also referred to as the network Q) increases, these two effects can make an L network impractical — particularly at transmitter power levels.

Pi and T networks are used to convert the impedances more gradually, putting less stress on the

More Matching Material

We've barely scratched the surface of impedance matching techniques — a topic widely discussed in amateur radio. The ARRL has published many articles over the years but one of the best, "Another Look At Reflections," was written by Walt Maxwell W2DU. This multi-part article is available online at www.arrl.org/transmission-lines along with numerous other interesting and educational articles.

components. The resulting impedance match is made over a wider frequency range, too, requiring fewer adjustments as frequency changes.

Pi networks with series inductors are very common in HF (high frequency, 3-30 MHz) transmitter outputs because the series inductor acts as a filter to reduce higher frequency harmonics from the transmitter. An L-C network with inductors in the series signal path between input and output is called "low pass" because of the additional attenuation by the inductors at higher signal frequencies. This is useful in meeting FCC (Federal Communications Commission) specifications for transmitter output harmonics and other spurious emissions.

T networks are the most common type of circuit used for adjustable impedance matching units. These are high pass networks because of the series capacitance in the signal path. Even though these circuits don't attenuate harmonics, the series capacitors make them less expensive to build and so they are popular in amateur equipment. Note that both the pi and T networks can be built with the opposite configuration of L and C, but those in the **figure** are the most common in use today for a variety of practical reasons.

Transmission Lines as Transformers

Transmission lines can be used to create reflections that cancel the unwanted reflections. Called synchronous transformers, the two best-known designs are shown in **Figure 7**. The quarter-wave transformer (or Q section) works by inserting a length of transmission line between Z_1 and Z_2 that is one-quarter of a wavelength (λ) long at the frequency of the impedance match. The characteristic impedance of the matching section, Z_Q , should be the geometric mean of Z_1 and Z_2 as shown in the **figure**.

For example, a one wavelength loop has a feed point impedance of around 120Ω . The geometric mean of 120Ω and 50Ω is 77.5Ω — quite close to the 75Ω impedance of RG-6, RG-59, or RG-11. There are many other types of coax with useful impedances, as well.

The Q section is found in many more places than just coaxial cable! If you wear glare-reducing coated glasses, the coating is a stack of transparent films. Each film is $1/4\lambda$ thick over a range of light wavelengths, and has a

characteristic impedance such that the result is a set of canceling reflections created for a variety of wavelengths of visible light.

A similar trick is employed in the 12th wave section. Surplus CATV hardline is free or very cheap, but is usually 75Ω cable which would create a 1.5:1 SWR in a 50Ω system. By inserting two back-to-back $1/12^{\text{th}}$ wavelength sections of the 75Ω and 50Ω cable as shown in the **figure**, reflections from the 75Ω cable are cancelled, making the system "look like" 50Ω — usually across an entire band of frequencies.

Are We Matched Yet?

I've presented four common techniques for impedance matching in this column: resistive, transformers, reactances, and transmission lines. You might recognize these in circuits you've seen on your workbench. Or, you might find one of these techniques to be just the thing you need to fix a pesky SWR problem or clean up some digital signals on a PCB. What's important is to remember that once the frequencies of signals start exceeding a few hundred kHz, everything starts behaving like an RF signal — because it is! **NV**

WORLD'S MOST VERSATILE
CIRCUIT
BOARD HOLDERS

Model 324

Our Circuit Board Holders add versatility & precision to your DIY electronics project. Solder, assemble & organize with ease.

VISIT US ON
Facebook Twitter YouTube

MONTHLY CONTEST
Visit us on Facebook® to post a photo of your creative PanaVise project for a chance to win a PanaVise prize package.

Model 201

PANAVISE®
Innovative Holding Solutions

7540 Colbert Drive • Reno • Nevada 89511 | (800) 759-7535 | www.PanaVise.com

>>> QUESTIONS

Don't Burn The Pi

I'm working on a project that requires putting a Raspberry Pi board into a small waterproof box. There is no air circulation for cooling. Is it likely the board will overheat and destroy itself? If so, how can I prevent a meltdown? I'm thinking of using thermal grease and pressing the processor heatsink against the enclosure. Could this work? Any other suggestions?

#3161

Samuel Ortiz
Dallas, TX

Snap, Crackle, POP

Is there a method or modification to prevent the loud pop when removing the headphones from the jack on my tube audio system?

I tried a capacitor across the jack to minimize the pop which worked, but at the expense of all the highs in the music. I'm kind of an audiophile type, so a reduction in sound quality is unacceptable to me. Anyone have a better solution to try?

#3162

Andrew Slater
Saginaw, MI

>>> ANSWERS

[#11152 - November 2015] Controller Quandary

I'm stuck with the limitations of my controller which has eight analog inputs each which can sense 0.5V changes between 0-10V. I need to measure temperature between 50-160°F within one or two degrees.

My idea to get the accuracy

needed is to divide the thermistor output across three inputs where input one would resolve the 100's, input two would resolve the 10's, and three would be the 1's. Example: temp 143 divided into three would produce a 1V signal on input one, a 4V signal on input two, and a 3V signal on input three. Then, in software in the controller, recombine the values back into a single temperature.

Does this concept seem doable and if yes, what would be the easiest way to create such a circuit? Thanks in advance for any assistance you can provide.

#1 You don't say what kind of controller you are using but if it has SPI capability, why not use a DS18B20 serial sensor. It is accurate to .5°C in the temperature range you specified and software routines for most MCUs are readily available.

Gene Sellier
Fairhope, AL

#2 Have you considered a digital sensor, such as the DS18B20? They are inexpensive (about \$4), readily available, accurate to $\pm 0.5^\circ\text{C}$ (so, about 1°F), have a wide temperature range, and then you only need one digital pin. I used one in a little oven controller that used a PIC a while back. Programming is a little more work, and it can be a little tricky to use with longer cables.

Jay Jaeger
Madison, WI

#3 I suggest you consider the Maxim MAX31820 temperature sensor. Its accuracy is $\pm 3-6^\circ\text{F}$ over the

range of -67°F to 257°F . It has a digital output (rather than analog) with a "1-wire" interface using only one controller port pin. A large number of these sensors can be paralleled on the one pin, so you could even put several together and average the readings. For a datasheet, see <https://datasheets.maximintegrated.com/en/ds/MAX31820.pdf>

Mark Peterson
Plymouth, MN

#4 I think a couple of quad op-amps would enable you to scale the temperature range you want to zero to five volts. That way, the resolution would be around 0.5 degrees on each A/D input.

John McCullough
La Habra, CA

[#2163 - February 2016] Cranky Flashlight

I have a three-LED hand-crank flashlight. I've included a copy of the PCB (orange) hoping to find out where the problem may be. The switch (SW1) provides two modes of lighting: one LED only (LED2) and all LEDs (LED1-LED3); in the OFF mode, the hand-crank is to be used in order to recharge the 3.6V rechargeable battery.

The hand-crank flashlight works in either of the two modes ONLY WHEN the hand-crank is being used. When I stop hand-cranking it, that's when lighting stops. I assume the li-on battery is not being charged (no white stuff around it for poor connection).

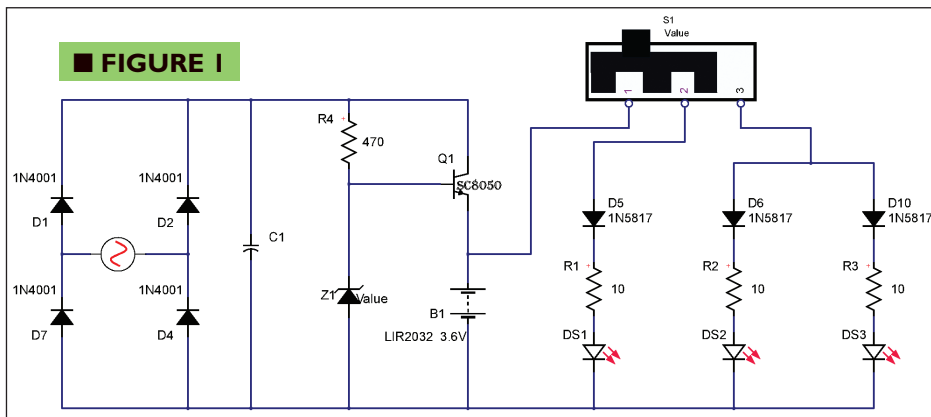
The black/plastic transistor is marked as SC8050. NTE (NTEinc.com) doesn't have a replacement for it. Can

All questions AND answers are submitted by Nuts & Volts readers and are intended to promote the exchange of ideas and provide assistance for solving technical problems. All submissions are subject to editing and will be published on a space available basis if deemed suitable by the publisher. Answers are submitted by readers and

NO GUARANTEES WHATSOEVER are made by the publisher. The implementation of any answer printed in this column may require varying degrees of technical experience and should only be attempted by qualified individuals.

Always use common sense and good judgment!

Send all questions and answers by email to forum@nutsvolts.com
or via the online form at www.nutsvolts.com/tech-forum



someone suggest a replacement and if it's NOT the transistor, suggest where the problem might be?

#1 Use a DMM to see if the li-ion battery is being charged. When the crank is being turning, the battery should have around 4.1 VDC across it, if the charging circuit is working correctly. A replacement for the 2SC8050 is the SS8050 (try www.mouser.com/Semiconductors/Discrete-Semiconductors/Transistors/Bipolar-Transistors-BJT/_/N-ax1sh?P=1z0z63xZ1z0z60l&Keyword=SS8050&FS=True). The li-ion battery may need to be replaced because it cannot hold a charge, so if you have the correct voltage across the battery with the crank turning, you probably need another battery.

Tim Brown
N&V Q&A

#2 I've drawn the schematic in **Figure 1** to show what I believe the design intent for this flashlight was. It is a little different than the presented board, but works to explain what is probably wrong.

Since the lights come on when cranked, the full wave rectifier section is working and the transistor is conducting. I believe the intent of the ZD1, R4, and Q1 is to maintain the voltage at the battery at a safe maximum level. This would occur

if ZD1 was actually a zener diode instead of a 1N4148. The zener diode would have a breakdown voltage somewhere between 4.3 and possibly as high as 5.1 volts. The battery would then be held below the zener voltage by the BE drop of the transistor, for the range of values mentioned (3.6 and 4.4 volts); 4.4 volts is really too high, but 5.1 volt zeners are cheap and widely available so the manufacturer may have cheated a bit.

This is not a very sophisticated charge control circuit. The spec sheet for the battery allows no more than 50 mA of charging current. Depending on the generator, this could easily be exceeded by this design. The most likely problem is a dead (not just uncharged) battery. The battery could be dead because of age or overcharging. The overcharging may not only be because of the unsophisticated circuit. A shorted transistor would make the problem worse, as would an open zener diode.

If the transistor is shorted,

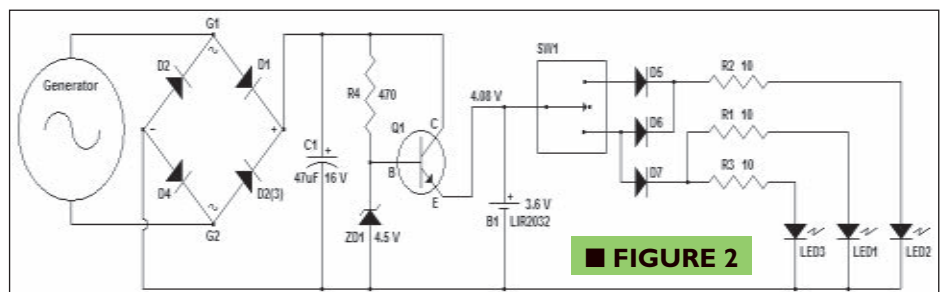
virtually any silicon NPN transistor will work as a replacement, as long as it can handle at least 50 mA. It will be easier to use if it has the same ECB arrangement of leads as the SC8050. A quick look through catalogs gives 2SC2655, TS13003, and APT27H as candidates.

Finally, there may not be anything wrong at all. I have similar flashlights, and when left unused for a long time it takes a lot of cranking to get the battery charged enough to light the lights. If the generator is specified to not exceed the battery limits too much, it should take almost half an hour to bring a completely discharged battery to full charge, and probably at least half that to get it to where it will light the LEDs.

Warren Wilderson
Shady Cove, OR

#3 The zener diode/transistor combination is a voltage regulator to provide an upper limit to the battery's charge voltage (see **Figure 2**). Somewhere on the Internet a few years back, (perhaps on YouTube), someone suggested replacing the failed battery with a 1 Farad memory back-up capacitor. At the time, Jameco carried a 1.5 Farad capacitor of the same physical size, and with some minor board modifications I was able to mount it in place of the battery. It now seems to work better than with the original battery.

Russ Wesp
via email



then all the way up the wheel train to the mainspring. I always think of the lowly hairspring as the "heartbeat" of a watch or clock. Similarly, when the hairspring gets magnetized, it is so small that the wildly changing N-S poles of the magnetic charge cause the poor thing to wind / unwind out of control, resulting in erratic timekeeping. Your fix, however, sounds much more elegant than mine.

Finally, whenever I have a watch / clock question, I always reach for the "bible:" *Handbook for Watch & Clock Repair* by H.G. Harris, 1961; Emerson Books. It's long been out of print, but this book is a valuable reference and worth looking for. The assembly illustrations alone are worth it.

Dave Prochnow

Excellent explanation — and reference suggestion. My latest adventure is rebuilding ATMOS clocks. Much simpler than rebuilding watches, but not necessarily easier. I've discovered there's a practical reason for the glass case. The slightest disturbance and it stops. So, no bookshelf clock. I don't have a fireplace, but I found a nice stable countertop. Works great there.

Bryan Bergeron

The Magic of Magnetism

Your November editorial on magnetism really stuck (pun intended). It reminded me that we don't think as much about magnetism today than we did years ago. We may even overlook that without magnetism, there would be no useful electricity. Wasn't magnetism one of the magic things that inspired us as kids? Who didn't wind a coil of wire around a nail and connect a battery? Who didn't play with iron filings and a magnet? Who didn't ponder why our nail electromagnet was eating up our batteries?

We find magnetism in motors and generators. They are in speakers, microphones, phonograph pick-ups, tape heads,

and disk drives. They are all over electro-mechanical devices like copiers and printers where they perform position sensing and even apply toner to the photoreceptor with a fine brush of magnetic developer. Many car engines have magnetic distributors, crankshaft sensors, and camshaft sensors. Let's not forget the simple things like the magnetos in lawn mower engines.

Then, there are the moments when your experience with magnetism saves the day. In the early 1990s, my friend bought an expensive Sony Trinitron CRT display for his computer. It was supposed to be the best of its day. He called me complaining that there was a problem with the screen image at the top right of the screen. It was wiggling ever so slightly. I asked him if he had any florescent lights near the monitor. No. Any other electronic device nearby? No. Just the old analog electric alarm clock on the shelf above the monitor. He moved the clock and the display was fine. Those old shaded pole clock motors were notorious for belting out a strong magnetic field!

Years later, I was in a "dollar store" checkout line. The cash register had a small green monochrome CRT screen and the display image was shaking wildly. It was summertime and the clerk had a personal fan near the register. I told him if he moved his fan the display would stop shaking. Another shaded-pole motor. He thought I was a genius!

No story-telling would be complete without mentioning demagnetism. A friend's child applied a powerful magnet to the front of their new computer CRT monitor. Of course, we know that creates severe color purity problems. Even the built-in degaussing was not helpful. The store told them they needed a new monitor. I went to their house with one of those big bulk tape demagnetizers. I started rotating the energized demagnetizer in the center of the screen and gradually worked my way away from the screen before

turning off the power. Good as new! Just like high voltage is useful but undesired when it is in the wrong place at the wrong time, magnetism can be a good friend and a frustrating enemy. Oh, and be careful of those powerful magnets used in hard drives. They are so strong they can pinch your skin and draw blood!

Rick Swenton
Bristol, CT

Wow, you've had quite an experience with incidental magnetism. My latest "gripe" or perhaps annoyance is an expensive watch winder that I picked up online. The specification didn't mention that the door was secured by magnets! Who in their right mind would put super magnets within inches of a rotating (expensive) mechanical watch for days at a time? Thanks for sharing.

Bryan Bergeron

CLASSIFIEDS

LIGHTING



LED lighting for displays, props, scenery, signs, models and more.

www.j2ledlighting.com



Automatic Stair Lighting

www.ReactiveLighting.com

Subscribe today!

Like what you see? Go to www.nutsvolts.com

HARDWARE WANTED

DEC EQUIPMENT WANTED!!!

Digital Equipment Corp. and compatibles.
Buy - Sell - Trade

CALL KEYWAYS 937-847-2300
or email buyer@keyways.com

ROBOTICS

Need sensors?



www.maxbotix.com

The EEL1 cable socket light assembly is optimized for 6 lb density foam board when an 11 mm (7/16") hole is used; it can be mounted into higher density boards with the hole size increased slightly to reduce insertion pressure, or with lower density foam board reducing the hole diameter for a tighter grip.

Socket taper dimensions are 12 mm (0.47") diameter typ. down to 9.7 mm (0.38") diameter typ. over a 33 mm length.

For mounting holes in HDU rigid foam board, make the hole 11 mm-0.25/+0.50 depending on the density and grip desired. For harder materials, a 12 mm (0.47") hole can be used with adhesive or resin to retain the socket. The smaller taper end of the socket has slots that can be used for resin back filling for additional retention in material over 25 mm (1") thick. In most applications using HDU as the mounting material, the press-fit grip from the socket taper



is sufficient to retain it.

The socket cable has a length of 254 mm (10") typ. with a female DC barrel connector jack 2.1 mm x 5.5 mm; jack diameter is 10.55 mm (0.415") typ.

The light source is a Candle Bi-Pin Lamp (CBL1) G4 type, output 30 lumens typ. 1,450 Kelvin. At 12 volts DC, 80 mA typ., 1.0 watts typ. Color coordinates are $x = 0.588$ / $y = 0.385$ ref.

The CBL1 is removable from the socket for insertion or replacement with a nominal height of one inch;

the overall cable length from the top of the lamp to the bottom of the barrel connector is 279.4 mm (11") typ. Price is \$15.99.

For more information, contact:
J2 LED Lighting, LLC
www.j2ledlighting.com

ANSWERS TO NAME THAT PART: 1a, 2a, 3c, 4b, 5a, 6c, 7b, 8b, 9b, 10a, 11c, 12a, 13c, 14c, 15a, 16c, 17b, 18c.

■ **LOOK FOR** ■ **SEARCH FOR** ■ **FIND**
Find your favorite advertisers here!

ADvertiser INDEX

AMATEUR RADIO AND TV

National RF25
Palstar33

BUYING ELECTRONIC SURPLUS

All Electronics Corp.36

CIRCUIT BOARDS

AP Circuits23
ExpressPCB7
Front Panel Express LLC22
Saelig Co. Inc.22

COMPILERS

CCS3
M.E. Labs32

COMPONENTS

All Electronics Corp.36
Electronix Express17
Qkits25

MICROCONTROLLERS / I/O BOARDS

Ion Motion Control23
M.E. Labs32
Technologic Systems45

DATA ACQUISITION

Measurement Computing2

DESIGN/ENGINEERING/REPAIR SERVICES

Cleveland Institute of Electronics..6
ExpressPCB7
Front Panel Express LLC22

National RF25

DEVELOPMENT PLATFORMS/TOOLS

AnarenBack Cover
CCS3
M.E. Labs32

EDUCATION

Cleveland Institute of Electronics..6
Command Productions37
NKC Electronics25
M.E. Labs32
Poscope7

EMBEDDED SYSTEMS

ImageCraft55

ENCLOSURES

Front Panel Express LLC22
Probotix37

EVENTS

Maker Faire66
RoboGames47

KITS & PLANS

National RF25
NKC Electronics25
Qkits25

MISC./SURPLUS

All Electronics Corp.36

MOTORS / MOTOR CONTROL

Ion Motion Control23
Qkits25

Robot Power25

PROGRAMMERS

CCS3
ImageCraft55
M.E. Labs32

RF TRANSMITTERS/RECEIVERS

National RF25

ROBOTICS

Cleveland Institute of Electronics..6
Electronix Express17
Ion Motion Control23
Robot Power25

TEST EQUIPMENT

NKC Electronics25
Pico Technology67
Poscope7
Saelig Co. Inc.22

TOOLS

PanaVise61
Poscope7
Probotix37

WIRE, CABLE AND CONNECTORS

All Electronics Corp.36

WIRELESS PRODUCTS

AnarenBack Cover
Technologic Systems45

All Electronics Corp.36

AnarenBack Cover

AP Circuits23

CCS3

Cleveland Institute of

Electronics6

Command Productions37

Electronix Express17

ExpressPCB7

Front Panel Express LLC ...22

ImageCraft55

Ion Motion Control23

Maker Faire66

Measurement Computing2

M.E. Labs32

National RF25

NKC Electronics25

Palstar33

PanaVise61

Pico Technology67

Poscope7

Probotix37

Qkits25

RoboGames47

Robot Power25

Saelig Co. Inc.22

Technologic Systems45

PRESENTED BY 

11th

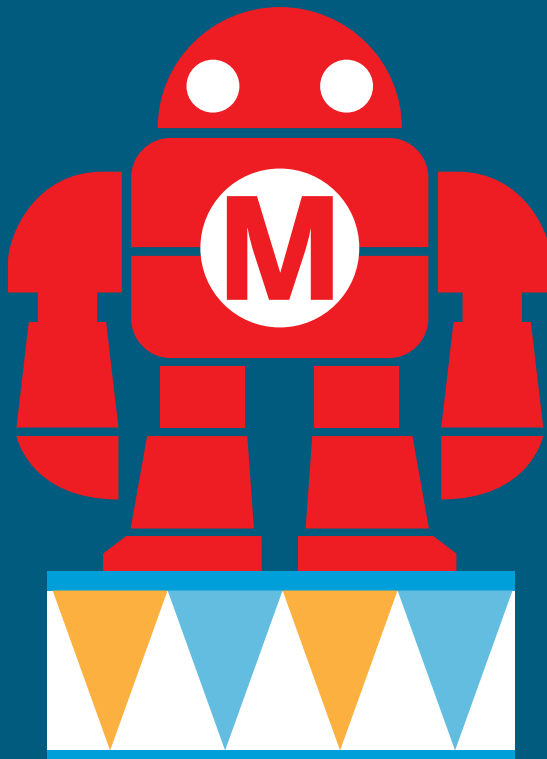
ANNUAL

Maker Faire®

BAY AREA

GREATEST

Show
& TELL
ON
Earth



MAY 20-22 SAN MATEO
EVENT CENTER

7TH ANNUAL WORLD MAKER FAIRE NEW YORK

OCT 1+2 NEW YORK
HALL OF SCIENCE

GET YOUR
TICKETS
TODAY!

makerfaire.com

BROUGHT TO YOU BY
Make: MAGAZINE



PicoScope 6.11 is now released

UPGRADE NOW

www.picotech.com/downloads

ALL FOR FREE

Users of all PicoScope versions old and new can upgrade to the latest software for free, for the life of their product.

You can also download the full software to try it for yourself in demo mode.

PicoScope Good News

GREAT NEW FEATURES INCLUDING:

- 16 serial protocols supported as standard:

Automotive: CAN, FlexRay, LIN, SENT **Avionics:** ARINC 429

Computer: Ethernet 10Base-T, 100Base-TX, PS/2, UART (RS-232, RS-442, RS-485), USB (FS, HS)

Embedded systems: 1-Wire, I2C, I2S, SPI **Lighting:** DMX512 **Hobby:** DCC

- Touchscreen functions • Mathematical waveform processing tools
- Frequency and duty-cycle v time plotting • Advanced waveform mathematics now includes user-configurable filters: High Pass, Low Pass, Band Pass and Band Stop

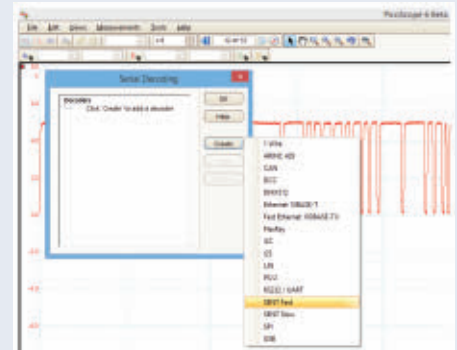
SENT focus

Background

SENT (Single Edge Nibble Transmission) is a serial interface originally designed for automotive applications. Lower cost than other serial protocols, SENT has become popular in high-resolution sensor applications such as throttle position, pressure, mass airflow, and temperature. Data is normally transmitted as two 12-bit data words in a message frame. The basic unit of time for SENT is a "tick". Each message frame starts with a synchronisation pulse of 56 ticks. Data is transmitted as nibbles - 4 bits of data encoded in the timing of successive falling edges. Nibble time encodes the data in the measured number of tick units. 12 ticks duration = binary 0000 (Hex 0), 13 ticks = binary 0001 (Hex 1), 14 ticks = binary 0010 (Hex 2) and so on up to 27 ticks = 1111 (Hex F). The message frame ends with a CRC/checksum nibble and optional pause pulse.

SENT decoding with PicoScope

The first step is to acquire the SENT signal of interest using PicoScope. Then select Serial Decoding from the Tools menu. Click Create and select SENT from the list of available protocols. In the SENT configuration dialog select the PicoScope Data input channel, Tick Time, Sensor Type, and other parameters as necessary. Click OK to see the decoded SENT messages in the PicoScope graph display. If you check the In Table box, PicoScope will display SENT messages in a tabular listing format. Double-click a message in the table to see the same message in the graph display.



Oscilloscopes

Low Cost



Workhorse



High Channel Count



Flexible Resolution



High Sampling Rate



High Bandwidth



**VIEW OUR ENTIRE PC BASED OSCILLOSCOPE RANGE ONLINE AT
WWW.PICOTECH.COM OR CALL US TODAY AT 1.800.591.2796**

Anaren®

What'll we think of next?®



THINK



BUILD



CONNECT

Achieve wireless mobile connectivity for embedded devices with AIR for WICED



No matter where you look, chances are you'll find an embedded Anaren AIR product. Smart products and applications for people at work or play. Find out how you can develop the next embedded product and mobile app – on Anaren's free Atmosphere cloud platform. It's as easy as Think. Build. Connect.



www.anaren.com/airforwiced

**Anaren
will be at
Embedded
World 2016**

Stop by booth 5-144
for a peek of what we
are rolling out, live demos,
hands on workshops, and more!